SIEMENS

SIMOTION

SIMOTION SCOUT Kommunikation

Systemhandbuch

05/2009

Rechtliche Hinweise

Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

⚠GEFAHR
bedeutet, dass Tod oder schwere Körperverletzung eintreten **wird**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

⚠WARNUNG
bedeutet, dass Tod oder schwere Körperverletzung eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

⚠VORSICHT
mit Warndreieck bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

VORSICHT
ohne Warndreieck bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG
bedeutet, dass ein unerwünschtes Ergebnis oder Zustand eintreten kann, wenn der entsprechende Hinweis nicht beachtet wird.

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zugehörige Gerät/System darf nur in Verbindung mit dieser Dokumentation eingerichtet und betrieben werden. Inbetriebsetzung und Betrieb eines Gerätes/Systems dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieser Dokumentation sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

⚠WARNUNG
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
DEUTSCHLAND

℗ 06/2009

Vorwort

Vorwort

Inhalt

Das vorliegende Dokument ist Bestandteil des **Dokumentationspaketes System- und Funktionsbeschreibungen**.

Gültigkeitsbereich

Dieses Handbuch ist gültig für SIMOTION SCOUT Produktstufe V4.1 SP4:

- SIMOTION SCOUT V4.1 SP4 (Engineering System der Produktfamilie SIMOTION),

Informationsblöcke des Handbuches

Das vorliegende Handbuch beschreibt die Kommunikationsmöglichkeiten von SIMOTION-Systemen.

- **Übersicht Kommunikationsfunktionen und -dienste**

  Allgemeine Informationen welche Kommunikationsmöglichkeiten SIMOTION hat.

- **PROFIdrive**

  Beschreibung des PROFIdrive Profils.

- **PROFIBUS**

  Informationen zur DPV1 Kommunikation und zur Einrichtung und Programmierung der Kommunikation zwischen SIMOTION- und SIMATIC-Geräten.

- **Ethernet Allgemein (TCP/IP- und UDP-Verbindungen)**

  Informationen zur Einrichtung und Programmierung der Ethernet-Kommunikation zwischen SIMOTION- und SIMATIC-Geräten.

- **PROFINET IO**

  Informationen zum Projektieren von PROFINET mit SIMOTION

- **Routing - Kommunikation über Netzwerkgrenzen**

  Allgemeine Informationen zu Routing

- **SIMOTION IT**

  Allgemeine Informationen welchen IT- und Web-Funktionen SIMOTION zur Verfügung stellt.

- **Index**

  Stichwortverzeichnis zum Finden der Informationen

SIMOTION Dokumentation

Einen Überblick zur SIMOTION Dokumentation erhalten Sie in einem separaten Literaturverzeichnis.

Diese Dokumentation ist als elektronische Dokumentation im Lieferumfang von SIMOTION SCOUT enthalten.

Die SIMOTION Dokumentation besteht aus 9 Dokumentationspaketen, die etwa 80 SIMOTION Dokumente und Dokumente zu zugehörigen Systemen (z. B. SINAMICS) enthalten.

Zur SIMOTION Produktstufe V4.1 SP4 stehen folgende Dokumentationspakete zur Verfügung:

- SIMOTION Engineering System Handhabung
- SIMOTION System- und Funktionsbeschreibungen
- SIMOTION Service und Diagnose
- SIMOTION Programmieren
- SIMOTION Programmieren - Referenzen
- SIMOTION C
- SIMOTION P350
- SIMOTION D4xx
- SIMOTION Ergänzende Dokumentation

Hotline und Internetadressen

Siemens Internet-Adresse

Ständig aktuelle Informationen zu den SIMOTION Produkten, Produkt Support, FAQs finden Sie im Internet unter:

- allgemeine Informationen:
  - **http://www.siemens.de/simotion** (deutsch)
  - **http://www.siemens.com/simotion** (international)
- Dokumentation downloaden
  Weiterführende Links für den Download von Dateien aus Service & Support.
  http://support.automation.siemens.com/WW/view/en/10805436
- Dokumentation auf Basis der Siemens Inhalte individuell zusammenstellen mit dem My Documentation Manager (MDM), siehe **http://www.siemens.com/mdm**

  Der My Documentation Manager bietet Ihnen eine Reihe von Features zur Erstellung Ihrer eigenen Dokumentation.
- FAQs
  Informationen zu FAQs (frequently asked questions) finden Sie über **http://support.automation.siemens.com/WW/view/en/10805436/133000**.

Weitere Unterstützung

Um Ihnen den Einstieg in die Arbeitsweise mit SIMOTION zu erleichtern, bieten wir Kurse an.

Wenden Sie sich dazu bitte an Ihr regionales Trainings-Center oder an das zentrale Trainings-Center in D-90027 Nürnberg

Informationen über das Trainingsangebot finden Sie unter

www.sitrain.com

Technical Support

Bei technischen Fragen wenden Sie sich bitte an folgende Hotline:

	Europa / Afrika
Telefon	+49 180 5050 222 (gebührenpflichtig)
Fax	+49 180 5050 223
0,14 €/Min. aus dem deutschen Festnetz, abweichende Mobilfunkpreise möglich.	
Internet	http://www.siemens.com/automation/support-request

	Amerika
Telefon	+1 423 262 2522
Fax	+1 423 262 2200
E-Mail	mailto:techsupport.sea@siemens.com

	Asien / Pazifik
Telefon	+86 1064 757575
Fax	+86 1064 747474
E-Mail	mailto:support.asia.automation@siemens.com

Hinweis

Landesspezifische Telefonnummern für technische Beratung finden Sie im Internet:

http://www.automation.siemens.com/partner

Fragen zur Dokumentation

Bei Fragen zur Dokumentation (Anregungen, Korrekturen) senden Sie bitte ein Fax oder eine E-Mail an folgende Adresse:

Fax	+49 9131- 98 2176
E-Mail	mailto:docu.motioncontrol@siemens.com

Inhaltsverzeichnis

Einleitung

1

1.1 Thema Kommunikation in der SIMOTION Dokumentation

Übersicht

Zum Thema Kommunikation finden Sie Hinweise in den einzelnen Gerätehandbüchern, in den Programmierhandbüchern und in diesem Handbuch Kommunikation.

Handbuch Kommunikation

Das Handbuch Kommunikation behandelt insbesondere Informationen, die für eine Kommunikation der SIMOTION Geräte mit Geräten außerhalb der SIMOTION Familie wichtig sind, insbesondere zur SIMATIC.

In diesem Handbuch finden Sie deshalb die Erläuterung der notwendigen Projektierungsschritte, die auf beiden Seiten der Kommunikationspartner durchgeführt werden müssen um eine fehlerfrei funktionierende Kommunikationsbeziehung zu erhalten.

Deshalb wird in diesem Handbuch auch sehr intensiv auf die Einstellungen und die Programmierung der SIMATIC S7 Stationen als Kommunikationspartner der SIMOTION eingegangen.

Gerätehandbücher und Programmierhandbücher

Die Gerätehandbücher behandeln das Thema Kommunikation insbesondere aus der Sicht der Geräte selbst, d.h. bzgl. der elektrischen Eigenschaften der vorhandenen Schnittstellen, sowie deren Einstellmöglichkeiten mit dem Engineering-System SIMOTION SCOUT.

Weitere Informationen finden Sie auch in den Handbüchern **Modulare Maschinenkonzepte** und **Basisfunktionen**, die Teil des SIMOTION Dokumentationspaketes sind.

Hier finden Sie keine Hinweise, wie Ihre Partnerstationen einzustellen sind.

Übersicht Kommunikations-Funktionen und -Dienste 2

2.1 Netzwerkmöglichkeiten

2.1.1 Einleitung

Als integraler Bestandteil der "Totally Integrated Automation" (TIA) bieten die SIMOTION und SIMATIC Netzwerklösungen die für die Kommunikationsanforderungen Ihrer Anwendung erforderliche Flexibilität und Leistungscharakteristik, ganz gleich wie einfach oder komplex Ihre Anwendung sein mag.

Hinweis

Dieser Abschnitt. beschreibt allgemein, welche Kommunikations- Funktionen und Dienste es innerhalb der Siemens Automatisierungstechnik gibt. Das bedeutet nicht zwangsläufig, dass auch alle genannten Funktionen für SIMOTION zur Verfügung stehen. Die Details zu den von SIMOTION unterstützten Funktionen finden Sie in den Kapiteln 4 - 8.

SIMOTION und SIMATIC Netzwerke für jede Anwendung

Die SIMOTION Produkte unterstützen eine Vielzahl von Netzwerkmöglichkeiten. Mit diesen Netzwerklösungen können Sie die SIMOTION Geräte entsprechend den Anforderungen Ihrer Anwendung kombinieren.

Zur weiteren Optimierung der Netzwerklösungen bieten SIMOTION Produkte integrierte Kommunikationsdienste bzw. -funktionen für die Erweiterung der Leistungsfähigkeit des Netzwerkprotokolls.

2.1.2 PROFINET

Übersicht

PROFINET basiert auf dem offenen Standard Industrial Ethernet für die Industrieautomatisierung zur unternehmensweiten Kommunikation und erweitert die Fähigkeit zum Datenaustausch Ihrer Automatisierungskomponenten bis in die Office-Umgebung, so dass Sie Ihre Automatisierungskomponenten, sogar die dezentralen Feldgeräte und Antriebe, an Ihr Local Area Network (LAN) anschließen können.

Weil PROFINET alle Ebenen in Ihrer Organisation miteinander verbindet – von den Feldgeräten bis zu den Managementsystemen – können Sie das anlagenweite Engineering mittels herkömmlichen IT-Standards umsetzen. Wie bei jeder auf Industrial Ethernet basierenden Lösung unterstützt PROFINET elektrische, optische und drahtlose Netzwerke.

Weil PROFINET auf Industrial Ethernet basiert und es sich nicht um eine abgewandelte "PROFIBUS for Ethernet"-Implementierung handelt, kann PROFINET die vorhandene installierte Basis von Ethernet-kompatiblen Geräten ausnutzen. Auch wenn PROFINET kein Master/Slave-System ist, bieten die Kommunikationsdienste PROFINET IO und PROFINET CBA die von Automatisierungssystemen geforderte Funktionalität:

- Mit PROFINET IO können Sie dezentrale Feldgeräte (z.B. digitale oder analoge Signalbaugruppen) und Antriebe direkt an ein Industrial Ethernet-Subnetz anschließen.
- PROFINET CBA (Component Based Automation) unterstützt modulare Lösungen für die Maschinen- und Anlagenkonstruktion. Sie definieren Ihr Automatisierungssystem in Form von autonomen Komponenten, wobei jede Komponente aus unabhängigen und in sich geschlossenen Aufgabenstellungen besteht.

Beide Kommunikationsdienste bieten Echtzeitfunktionalität, um PROFINET zu einer Echtzeitimplementierung zu machen. Außerdem erlaubt PROFINET das gleichzeitige Vorhandensein der Echtzeitkommunikation Ihres Automatisierungsprozesses und Ihrer sonstigen IT-Kommunikation, zur gleichen Zeit im selben Netzwerk, ohne dass dadurch das Echtzeitverhalten Ihres Automatisierungssystems beeinträchtigt wird.

Zur weiteren Unterstützung von fehlersicheren bzw. "sicherheitsrelevanten" Anwendungen kommuniziert das PROFIsafe-Profil mit den fehlersicheren Geräten über das PROFINET-Subnetz.

2.1.3 Industrial Ethernet

Übersicht

Dadurch, dass Industrial Ethernet ein Kommunikationsnetzwerk für die Verbindung von Befehlsebene und Zellenebene bereitstellt, können Sie mit Industrial Ethernet die Fähigkeiten für den Datenaustausch Ihrer Automatisierungskomponenten in die Office-Umgebung ausdehnen.

Industrial Ethernet basiert auf den Normen IEEE 802.3 und IEEE 802.3u für die Kommunikation zwischen Computern und Automatisierungssystemen und ermöglicht es Ihrem System dadurch, große Datenmengen über lange Entfernungen auszutauschen.

2.1.4 PROFIBUS

Übersicht

PROFIBUS basiert auf den Normen IEC 61158 / EN 50170 und bietet eine Lösung mit offenem Feldbus für die komplette Produktions- und Prozessautomatisierung. PROFIBUS bietet schnellen und zuverlässigen Datenaustausch und integrierte Diagnosefähigkeiten. PROFIBUS unterstützt herstellerunabhängige Lösungen mit dem weltweit größten Fremdhersteller-Support. Für Ihr PROFIBUS-Subnetz können Sie eine Vielzahl von Übertragungsmedien nutzen: elektrisch, optisch und drahtlos.

PROFIBUS umfasst die folgenden Kommunikationsdienste:

- PROFIBUS DP (Decentralized Peripherals) ist ein Kommunikationsprotokoll, das sich besonders gut für die Produktionsautomatisierung eignet.
  PROFIBUS DP bietet schnellen, zyklischen und deterministischen Austausch von Prozessdaten zwischen einem Bus-DP-Master und den zugeordneten DP-Slavegeräten. PROFIBUS DP unterstützt die isochrone Kommunikation. Die synchronisierten Ausführungszyklen stellen sicher, dass die Daten in konsistenten äquidistanten Zeitabständen übertragen werden.
- PROFIBUS PA (Process Automation) erweitert PROFIBUS DP und bietet eigensichere Daten- und Leistungsübertragung entsprechend der Norm IEC 61158-2.
- PROFIBUS FMS (Fieldbus Message Specification) ist für die Kommunikation auf Zellebene ausgelegt, wo die Steuerungen miteinander kommunizieren. Mittels PROFIBUS FMS können Automatisierungssysteme verschiedener Hersteller miteinander kommunizieren.
- PROFIBUS FDL (Fieldbus Data Link) wurde für die Übertragung von mittleren Datenmengen optimiert, um die fehlerfreie Übertragung von Daten auf dem PROFIBUS Subnetz zu unterstützen.

Außerdem nutzt PROFIBUS Profile, um Kommunikationsmöglichkeiten für die Anforderungen bestimmter Anwendungen zu bieten, z.B. PROFIdrive (für die Bewegungssteuerung) oder PROFIsafe (für fehlersichere bzw. "sicherheitsrelevante" Anwendungen).

2.1.5 MPI (mehrpunktfähige Schnittstelle)

Übersicht

MPI sind integrierte Schnittstellen für SIMOTION und SIMATIC Produkte (SIMOTION Geräte, SIMATIC S7 Geräte, SIMATIC HMI sowie SIMATIC PC und PG).

MPI bietet eine Schnittstelle zur PG/OP-Kommunikation. MPI bietet auch einfache Netzwerkfähigkeit unter Verwendung der folgenden Dienste: Kommunikation über globale Daten (GD), S7-Kommunikation und S7-Basiskommunikation.

Das elektrische Übertragungsmedium für MPI nutzt die Norm RS 485, die auch von PROFIBUS verwendet wird.

2.1.6 Punkt-zu-Punkt-Kommunikation (PtP)

Übersicht

SIMOTION Geräte können programmiert werden, damit sie Daten mit einer anderen Steuerung im Netzwerk austauschen. Auch wenn die Punkt-zu-Punkt-Kommunikation nicht als Subnetz eingestuft ist, bietet die Punkt-zu-Punkt-Verbindung die serielle Übertragung, z.B. über RS232 oder RS485, von Daten zwischen zwei Stationen, z.B. mit einer SIMATIC Steuerung oder sogar mit einem Fremdgerät, das kommunikationsfähig ist.

Für die Punkt-zu-Punkt-Kommunikation können Sie CP-Baugruppen (z.B. eine CP340) oder ET200-Baugruppen verwenden, um Daten zwischen zwei Steuerungen zu lesen und zu schreiben. Die Punkt-zu-Punkt- Kommunikation stellt damit eine leistungsstarke und kostengünstige Alternative zu Buslösungen dar, insbesondere dann, wenn nur wenige Geräte an das SIMOTION Gerät angeschlossen werden sollen.

Die Punkt-zu-Punkt-Kommunikation bietet die folgenden Fähigkeiten:

- Anpassung an das Protokoll des Kommunikationspartners mittels Standardverfahren oder ladbaren Treibern.
- Definieren eines benutzerspezifischen Verfahrens mittels ASCII-Zeichen
- Kommunikation mit anderen Arten von Geräten wie Bedienstationen, Druckern oder Kartenlesegeräten

Weitere Literatur

Weiterführende Literatur zur Punk-zu-Punkt-Kommunikation finden Sie in den Beschreibungen der CP- bzw. ET200-Baugruppen.

2.2 Kommunikationsdienste (oder Netzwerkfunktionen)

2.2.1 Einleitung

SIMOTION und SIMATIC Geräte unterstützen einen Satz spezifischer Kommunikationsdienste, die die Datenpakete steuern, die über die physikalischen Netzwerke übertragen werden. Jeder Kommunikationsdienst definiert einen Satz Funktionen und Leistungsmerkmale, z.B. die zu übertragenden Daten, die zu steuernden Geräte, die zu beobachtenden Geräte und die zu ladenden Programme.

Kommunikationsdienste der SIMOTION und SIMATIC Produkte

Kommunikationsdienste, die auch häufig als Netzwerkfunktionen bezeichnet werden, sind die Softwarekomponenten, die die physikalische Hardware der Netzwerke nutzen. Softwareschnittstellen (z.B. S7-Systemfunktionen) im Endgerät (z.B. SIMOTION Gerät, SIMATIC S7 Gerät oder PC) bieten Zugriff auf die Kommunikationsdienste. Eine Softwareschnittstelle verfügt jedoch nicht unbedingt über alle Kommunikationsfunktionen für den Kommunikationsdienst. Ein solcher Dienst kann im jeweiligen Endsystem mit unterschiedlichen Softwareschnittstellen zur Verfügung gestellt werden.

2.2.2 PG/OP-Kommunikationsdienste

Übersicht

PG/OP-Dienste sind die integrierten Kommunikationsfunktionen, mit denen SIMATIC und SIMOTION Automatisierungssysteme mit einem Programmiergerät (z.B. STEP 7) und Bedien- und Beobachtungsgeräten kommunizieren. Alle SIMOTION und SIMATIC Netzwerke unterstützen die PG/OP-Kommunikationsdienste.

2.2.3 S7-Kommunikationsdienste

Übersicht

S7-Kommunikationsdienste bieten Datenaustausch mittels Kommunikations-Systemfunktionsbausteinen (SFBs) und Kommunikations-Funktionsbausteinen (FBs) für konfigurierte S7-Verbindungen.

Alle SIMOTION Geräte und SIMATIC S7 Geräte haben integrierte S7-Kommunikationsdienste, mit denen das Anwenderprogramm in der Steuerung das Lesen oder Schreiben von Daten auslösen kann. Diese Funktionen sind von spezifischen Netzwerken unabhängig, so dass Sie die S7-Kommunikation über jedes Netzwerk (MPI, PROFIBUS, PROFINET oder Industrial Ethernet) programmieren können.

Für die Übertragung von Daten zwischen den Steuerungen müssen Sie eine Verbindung zwischen den beiden Steuerungen konfigurieren. Die integrierten Kommunikationsfunktionen werden vom SFB/FB in der Anwendung aufgerufen. Sie können bis zu 64 KB Daten zwischen SIMOTION und SIMATIC S7 Geräten übertragen.

Sie können mit Ihrem HMI-Gerät, Programmiergerät (PG) oder Computer auf die Daten in der Steuerung zugreifen, weil die S7-Kommunikationsdienste in das Betriebssystem der SIMOTION Geräte und SIMATIC S7 Gerät integriert sind. Diese Art von Peer-to-Peer-Verbindung benötigt keine zusätzliche Verbindungseinrichtung. (Wenn Sie jedoch eine Verbindung zu einem dieser Geräte konfigurieren, können Sie über die symbolischen Namen auf die Daten zugreifen.)

Hinweis

SFBs können bei SIMOTION nicht verwendet werden.

2.2.4 S7-Basiskommunikationsdienste

Übersicht

S7-Basiskommunikationsdienste bieten Datenaustausch mittels Kommunikations-Systemfunktionen (SFCs) für nicht konfigurierte S7-Verbindungen. Diese SFCs (z.B. X_GET oder X_PUT) lesen oder schreiben die Daten auf ein SIMATIC Steuerung, um so kleine Datenmengen über ein MPI-Subnetz an eine andere S7-Station (S7-Steuerung, HMI oder PC) zu übertragen.

Die SFCs für die S7-Basiskommunikation kommunizieren nicht mit Stationen in anderen Subnetzen. Für die S7-Basiskommunikation brauchen Sie keine Verbindungen zu konfigurieren. Die Verbindungen werden aufgebaut, wenn das Anwenderprogramm die SFC aufruft.

Hinweis

Die S7-Basiskommunikationsdienste können Sie nur über eine MPI-Verbindung zwischen SIMATIC S7-300, S7-400 oder C7-600 Steuerungen nutzen.

2.2.5 Kommunikationsdienst "Globale Daten"

Übersicht

Neben den anderen Möglichkeiten für die Netzwerkkommunikation können Sie eine Kommunikationsverbindung 'Globale Daten' (GD) konfigurieren, um die zyklische Datenübertragung zwischen SIMATIC Steuerungen bereitzustellen, die an ein MPI-Netzwerk angeschlossen sind. Der Datenaustausch läuft im Rahmen des normalen Prozessabbildaustauschs ab, weil die globale Datenkommunikation in das Betriebssystem der SIMATIC Steuerung integriert ist.

Der Empfang der globalen Daten wird nicht quittiert, weil es sich bei der globalen Datenkommunikation um ein Verfahren zum Übertragen von Daten handelt. Ein Publisher (Datenquelle) sendet die Daten an einen oder mehrere Subscriber (Datensenke), und die Subscriber empfangen die Daten. Der Publisher empfängt keine Quittierung von den Subscribern darüber, dass diese die übertragenen Daten erhalten haben.

Hinweis

Die globale Datenkommunikation können Sie nur über eine MPI-Verbindung zwischen SIMATIC S7-300, S7-400 oder C7-600 Steuerungen nutzen.

Die GD-Kommunikation erfordert keine besondere Programmierung und auch keine Programmbausteine in Ihrem STEP 7-Anwenderprogramm. Die Betriebssysteme der einzelnen Steuerungen bearbeiten den Austausch von globalen Daten. Mit STEP 7 konfigurieren Sie eine globale Datentabelle (GD) mit dem Quellpfad der Daten, die an die Subscriber übertragen werden sollen. Diese GD-Tabelle wird mit der Hardware-Konfiguration für den Publisher und die Subscriber geladen.

Globale Daten sind für SIMOTION nicht verfügbar.

2.2.6 PROFINET-Kommunikationsdienste

Übersicht

PROFINET umfasst die folgenden Kommunikationsdienste:

- Mit dem Kommunikationsdienst PROFINET IO können Sie E/A-Geräte und Antriebe über Ethernet-Physik an die SIMOTION oder SIMATIC Steuerung anbinden. Mit PROFINET IO kann das in der Steuerung ausgeführte Anwenderprogramm die Eingangs- und Ausgangsdaten der E/A-Geräte und Antriebe verarbeiten. Die Adressierung für PROFINET IO konfigurieren Sie in STEP 7 bzw. SIMOTION SCOUT.
- Mit PROFINET CBA können Sie Ihr Automatisierungssystem nach autonomen Untereinheiten oder Komponenten definieren. Bei diesen Komponenten kann es sich um PROFINET IO-, PROFIBUS DP- oder auch Fremdgeräte und -subnetze handeln.

Wenn Sie die Kommunikationsdienste PROFINET CBA für eine komponentenbasierte Lösung einsetzen möchten, konfigurieren Sie die SIMATIC Steuerungen und die E/A-Geräte in einzelnen Komponenten in STEP 7. Anschließend konfigurieren Sie die Kommunikation zwischen den verschiedenen Komponenten mit SIMATIC iMAP.

Beide Kommunikationsdienste PROFINET IO und PROFINET CBA bieten die von Automatisierungssystemen geforderte Echtzeitkommunikation.

Hinweis

PROFINET CBA steht nur für SIMATIC-Geräte zur Verfügung, jedoch noch nicht für SIMOTION Geräte.

2.2.7 Industrial Ethernet-Kommunikationsdienste

Übersicht

Industrial Ethernet basiert auf den Normen IEEE 802.3 und IEEE 802.3u und verbindet die Automatisierungssysteme mit Ihrem Business-System, so dass Sie im Büro Zugang zu den Daten haben und diese bearbeiten können.

Industrial Ethernet umfasst die folgenden Kommunikationsdienste:

- Die ISO-Übertragung bietet Dienste zum Übermitteln von Daten über Verbindungen, die die fehlerfreie Datenübertragung unterstützen. ISO-Übertragung ist nur mit STEP7 möglich.
- Mit TCP/IP können Sie zwischen Steuerungen und Computern in PROFINET oder Industrial Ethernet-Netzwerken zusammenhängende Datenblöcke austauschen. Bei TCP/IP sendet die Steuerung zusammenhängende Datenblöcke.
- ISO-on-TCP (RFC 1006) unterstützt die fehlerfreie Datenübertragung. Bei SIMOTION nur, wenn über den SCOUT ONLINE gegangen wird. Falls die Kommunikation aus dem Anwenderprogramm erfolgt, muss RFC selbst programmiert werden.
- UDP (User Datagram Protocol) und UDP Multi-Cast bietet einfache Datenübertragung ohne Quittierung. Sie können zusammenhängende Datenblöcke von einer Station zu einer anderen übertragen, z.B. zwischen einer SIMOTION und SIMATIC Steuerung, einem PC oder einem Fremdsystem.
- Bei der IT-Kommunikation (Informationstechnologie) können Sie mittels Standard-Ethernet Protokollen und Diensten wie z.B. FTP, HTTP und E-Mail über PROFINET- oder Industrial Ethernet-Netzwerke Daten gemeinsam nutzen.

2.2.8 PROFIBUS-Kommunikationsdienste

Übersicht

PROFIBUS umfasst die folgenden Kommunikationsdienste:

- PROFIBUS DP (Distributed Peripherals) fördert die transparente Kommunikation mit der dezentralen Peripherie. Das SIMOTION/STEP 7-Anwenderprogramm greift genauso auf die dezentrale Peripherie zu wie auf die E/A im zentralen Baugruppenträger der Steuerung (bzw. der SPS). PROFIBUS DP ermöglicht die direkte Kommunikation mit der dezentralen Peripherie. PROFIBUS DP entspricht den Normen EN 61158 / EN 50170.
- PROFIBUS PA (Process Automation) erleichtert die direkte Kommunikation mit Instrumenten der Prozessautomatisierung (PA). Dies umfasst sowohl den zyklischen Zugriff auf die E/A, typischerweise mit einem SPS-Master, als auch den azyklischen Zugriff auf den potentiell umfangreichen Satz von Gerätebetriebsparametern, typischerweise mit einem Engineering-Werkzeug wie Process Device Manager (PDM). PROFIBUS PA entspricht der Norm IEC 61158
- PROFIBUS-FMS (Fieldbus Message Specification) ermöglicht die Übertragung von strukturierten Daten (FMS-Variablen). PROFIBUS FMS entspricht der Norm IEC 61784.

- PROFIBUS FDL (Fieldbus Data Link) wurde für die Übertragung von mittleren Datenmengen optimiert, um die fehlerfreie Übertragung von Daten auf dem PROFIBUS Subnetz zu unterstützen. PROFIBUS FDL unterstützt die SDA-Funktion (Daten mit Quittierung senden)

Hinweis

SIMOTION Geräte unterstützen ausschließlich den Kommunikationsdienst PROFIBUS DP.

Für die fehlersichere Kommunikation nutzen SIMOTION und SIMATIC Geräte das Profil PROFIsafe für PROFIBUS DP.

Für die Kommunikation zwischen SIMOTION Geräte hin zu den angeschlossenen Antrieben nutzen die SIMOTION Geräte das Profil PROFIdrive.

Weitere Literatur

Einen Gegenüberstellung der SIMATIC S7 und SIMOTION Systemfunktionen finden Sie im Verzeichnis 2_FAQ auf der Utilities & Applications CD.

2.3 Weitere Verfahren zum Austauschen von Informationen

Neben den Standardkommunikationsnetzwerken unterstützen SIMOTION und SIMATIC auch weitere Mittel zur gemeinsamen Nutzung von Informationen über Netzwerke.

Gemeinsame Datennutzung mit anderen Anwendungen über OPC (OLE for Process Control)

Mit OPC (OLE for Process Control) können Windows-Anwendungen auf Prozessdaten zugreifen, so dass auf einfache Weise Geräte und Anwendungen verschiedener Hersteller miteinander kombiniert werden können. OPC bietet nicht nur eine offene, herstellerunabhängige Schnittstelle, sondern auch eine bedienerfreundliche Client/Server-Konfiguration für den standardisierten Datenaustausch zwischen (z.B. HMI- oder Office-Anwendungen), die kein spezifisches Netzwerk oder Protokoll benötigen.

Der OPC-Server bietet Schnittstellen für den Anschluss der OPC-Client-Anwendungen. Sie konfigurieren die Client-Anwendung für den Zugriff auf Datenquellen, z.B. Adressen im Speicher einer SPS. Weil mehrere verschiedene OPC-Clients gleichzeitig auf denselben OPC-Server zugreifen können, können dieselben Datenquellen für jede OPC-konforme Anwendung genutzt werden.

Neben OPC-Servern bietet SIMATIC NET auch Anwendungen zum Konfigurieren und Testen von OPC-Verbindungen: Advanced PC Configuration (APC) und OPC Scout (dient zum Testen und in Betrieb nehmen einer OPC-Anwendung bzw. eines OPC-Servers). Mit diesen Werkzeugen verbinden Sie SIMOTION und SIMATIC S7-Produkte mit anderen OPC-konformen Anwendungen.

Die SIMATIC NET OPC-Server unterstützen die folgenden Kommunikationsdienste:

- PROFINET IO (mittels PROFINET- oder Industrial Ethernet-Subnetz)
- PROFINET CBA (mittels PROFINET- oder Industrial Ethernet-Subnetz)
- TCP/IP (mittels PROFINET- oder Industrial Ethernet-Subnetz)
- PROFIBUS DP (mittels PROBUS-Subnetz)

- PROFIBUS FMS (mittels PROFIBUS-Subnetz)
- S7-Kommunikation
- S5-kompatible Kommunikation

Gemeinsame Datennutzung in einer Office-Umgebung mittels Informationstechnologie (IT)

SIMOTION und SIMATIC nutzen die herkömmlichen IT-Werkzeuge (wie E-Mail - nur SIMATIC, HTTP-Webserver, FTP und SNMP) mit PROFINET und Industrial Ethernet-Netzwerken, um die Fähigkeiten zur gemeinsamen Datennutzung in der Office-Umgebung zu erweitern.

Für die SIMOTION Geräte werden die entsprechenden Funktionen über SIMOTION IT DIAG zur Verfügung gestellt, siehe SIMOTION IT Ethernet-basierende HMI- und Diagnosefunktionen.

PROFIdrive

3

3.1 Warum Profile

Profile in der Automatisierungstechnik legen für Geräte, Gerätefamilien oder gesamte Systeme bestimmte Eigenschaften und Verhaltensweisen so fest, dass dadurch deren weit gehende, eindeutige Charakterisierung erreicht wird. Nur Geräte mit herstellerübergreifend gleichem Profil können sich an einem Feldbus "interoperabel" verhalten und damit die Vorteile eines Feldbusses für den Anwender voll erschließen.

Profile sind von Herstellern und Anwendern getroffene Festlegungen (Spezifikationen) über bestimmte Eigenschaften, Leistungsmerkmale und Verhaltensweisen von Geräten und Systemen. Sie haben das Ziel, Geräte und Systeme, die auf Grund einer "profilgemäßen" Entwicklung zu einer Produktfamilie gehören, an einem Bus interoperabel und bis zu einem gewissen Grad austauschbar betreiben zu können.

Man unterscheidet bei den Profilen zwischen so genannten Applikationsprofilen (allgemeinen oder spezifischen) und Systemprofilen.

- Applikationsprofile beziehen sich vorrangig auf Geräte, in diesem Fall Antriebe, und enthalten sowohl eine vereinbarte Auswahl an Buskommunikation als auch an spezifischen Geräteanwendungen.
- Systemprofile beschreiben Klassen von Systemen unter Einschluss der Masterfunktionalität, Programminterfaces und Integrationsmittel.

PROFIdrive

Das Profil PROFIdrive gehört zu den spezifischen Applikationsprofilen. Es beschreibt im Detail die sinnvolle Anwendung der Kommunikationsfunktionen Querverkehr, Äquidistanz und Taktsynchronisierung in Antriebsapplikationen. Ferner werden alle Geräteeigenschaften, die Einfluss auf die Schnittstelle zu einem über PROFIBUS oder PROFINET verbundenen Controller haben, klar spezifiziert. Dazu gehören u.a. die State maschine (Ablaufsteuerung), das Geberinterface, die Normierung von Werten, die Definition von Standardtelegrammen, der Zugriff auf Antriebsparameter, die Antriebsdiagnose usw.

Das Profil PROFIdrive unterstützt dabei sowohl zentrale als auch dezentrale Motion Control Konzepte.

Die Grundphilosophie: – Keep it simple –

Das Profil PROFIdrive verfolgt die Grundphilosophie, dass die Antriebsschnittstelle so einfach wie möglich und frei von technologischen Funktionen gehalten wird. Durch diese Philosophie haben Referenzmodelle wie auch die Funktionalität und Performance des PROFIBUS-/PROFIRIVE-Masters keinen bzw. nur geringen Einfluss auf die Antriebsschnittstelle.

3.2 PROFIdrive Übersicht

Das PROFIdrive Profil

Das PROFIdrive Profil definiert das Geräteverhalten und das Zugriffsverfahren auf Antriebsdaten für elektrische Antriebe am PROFIBUS und am PROFINET, vom einfachen Freqenzumrichter bis hin zu hochperformanten Servoreglern.

PROFIdrive ist in einen allgemeinen Teil und einen busspezifischen Teil gegliedert.´Im allgemeinen Teil sind folgende Eigenschaften definiert:

- Basismodell (Base Model)
- Parametermodell (Parameter Model)
- Applikationsmodell (Application Model)

Im busspezifischen Teil finden die folgenden Zuordnungen statt:

- PROFIdrive auf PROFIBUS
- PROFIdrive auf PROFINET

Eine genaue Beschreibung des PROFIdrive Profils finden Sie unter nachfolgendem Hinweis.

Literatur-Hinweis

PROFIdrive Profil

PROFIBUS Profile PROFIdrive – Profile Drive Technology
Version V4.1, May 2006,
PROFIBUS User Organization e. V.
Haid-und-Neu-Straße 7, D-76131 Karlsruhe
http://www.profibus.com
Order Number 3.172, spez. Kap. 6

Normen

Norm IEC 61800

3.3 PROFIdrive Basis/Parameter Modell

Beschreibung

Das PROFIdrive Basismodell beschreibt ein Automatisierungssystem als eine Menge von Geräten sowie deren Beziehungen untereinander (Applikationsschnittstellen, Parameterzugriff). Folgende Geräteklassen werden im Basismodell unterschieden:

PROFIdrive	PROFIBUS DP	PROFINET IO
Controller (übergeordnete Steuerung oder Host des Automatisierungssystems)	DP Master Klasse 1	IO Controller
Peripheral Device (P-Device)	DP Slave (I-Slaves)	IO Device
Supervisor (Engineering Station)	DP Master Klasse 2	IO Supervisor

PROFIdrive Geräteklassen

Beispiel für ein PROFIdrive Automatisierungskonzept

Die nachfolgende Grafik zeigt ein typisches Automatisierungskonzept.

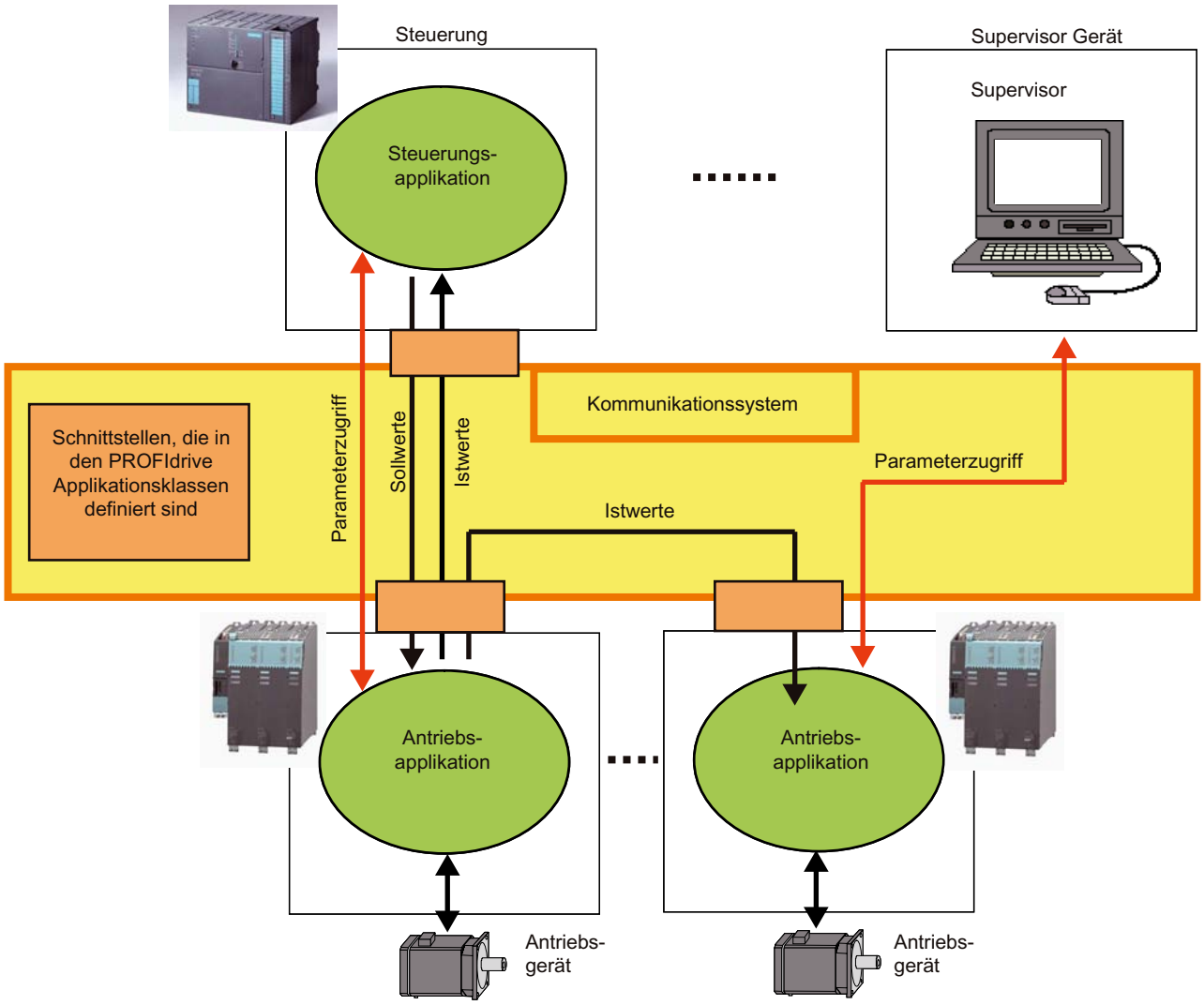


Bild 3-1 Automatisierungskonzept

Kommunikationsdienste

Im PROFIdrive Profil sind zwei Kommunikationsdienste definiert, der zyklische Datenaustausch und der azyklische Datenaustausch.

- Zyklischer Datenaustausch über zyklischen Datenkanal

  Motion Control Systeme benötigen im Betrieb zum Steuern und Regeln zyklisch aktualisierte Daten. Diese müssen über das Kommunikationssystem als Sollwerte an die Antriebsgeräte gesendet bzw. als Istwerte vom Antriebsgerät übertragen werden. Die Übertragung dieser Daten ist normalerweise zeitkritisch.

- Azyklischer Datenaustausch über azyklischen Datenkanal

  Neben dem zyklischen Datenaustausch steht außerdem ein azyklischer Parameterkanal zum Austausch von Parametern zwischen Steuerung / Supervisor und Antriebsgeräten zur Verfügung. Der Zugriff auf diese Daten ist nicht zeitkritisch.

- Alarmkanal

  Die Alarme werden ereignisgesteuert ausgegeben und zeigen das Kommen und Gehen von Fehlerzuständen an.

Die nachfolgende Grafik zeigen das Datenmodell und den Datenfluss im P-Device.

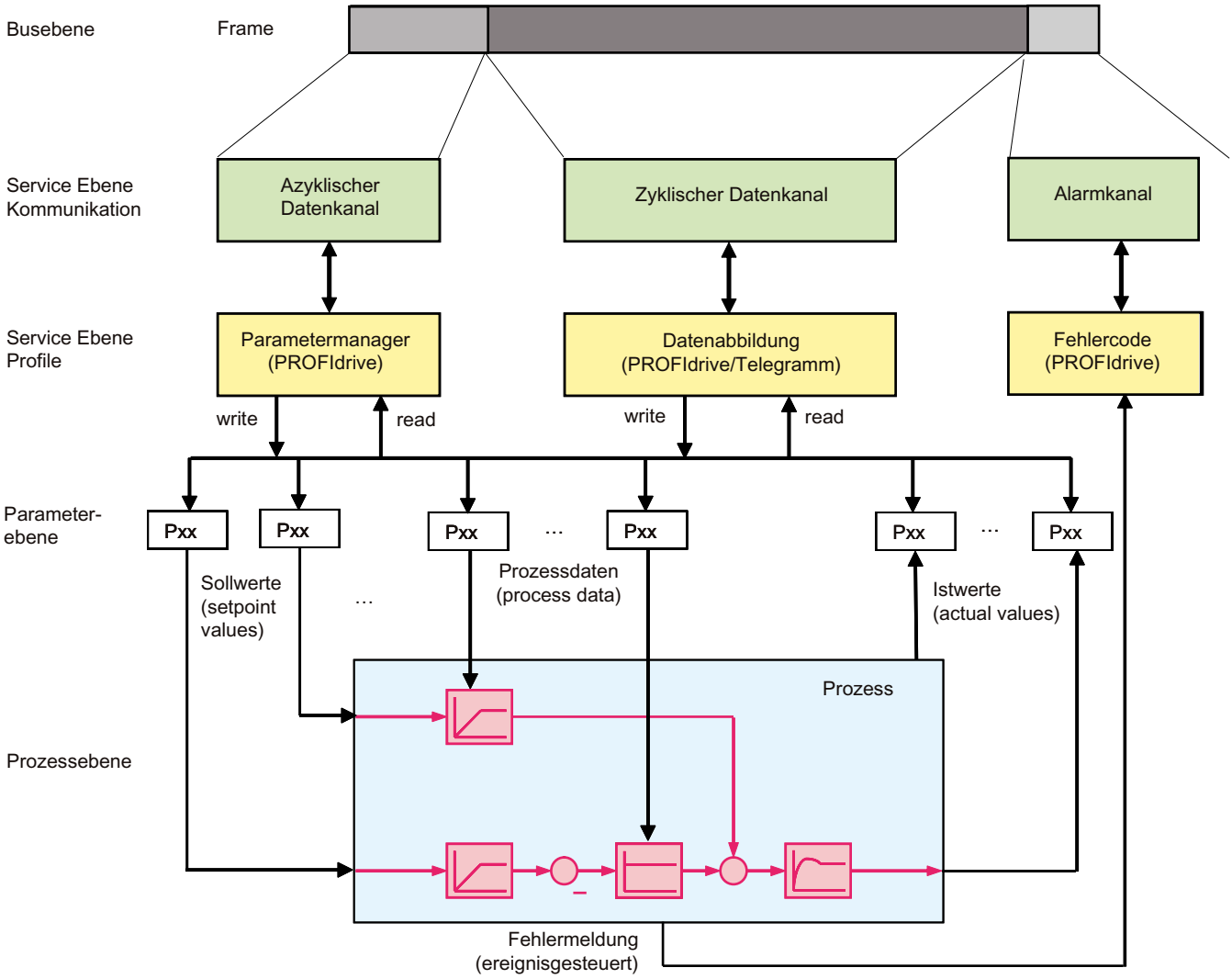


Bild 3-2 Datenmodell und Datenfluss im P-Device

Alarme und Fehlermeldungen

Die Alarme werden ereignisgesteuert ausgegeben und zeigen das Kommen und Gehen von Fehlerzuständen an.

Parameter Modell

Das Parameter Modell im PROFIdrive Profil unterscheidet zwischen den Profilparametern und den herstellerspezifischen Parametern:

- Die Profilparameter sind für Objekte definiert, die sich aus dem Gerätemodell des PROFIdrive Profils ableiten. Das sind z.B. allgemeine Funktionen wie Antriebsidentifikation, Störpuffer oder Antriebssteuerung. Diese Parameter sind für alle Antriebe gleich.

- Alle anderen Parameter sind herstellerspezifisch. Die Parameter werden nicht durch das Profil festgelegt, jedoch die Schnittstelle zum Applikationsprozess.

Der Zugriff auf die Elemente (Werte, Parameterbeschreibungen, Textelemente) eines Parameters erfolgt grundsätzlich azyklisch (mit Ausnahme von G120 Antrieben, die mit PKW zyklisch Daten austauschen können). Dazu ist eine unabhängige Request-/Response-Datenstruktur definiert.

3.4 Segmentierung in Applikationsklassen

Einbindung von Antrieben in Automatisierungslösungen

Die Einbindung von Antrieben in Automatisierungslösungen ist stark von der Antriebsaufgabe abhängig. Um die ganze, riesige Bandbreite an Antriebsanwendungen vom einfachsten Frequenzumrichter bis zu hochdynamischen, synchronisierten Mehrsachssystemen in einem Profil abdecken zu können, definiert PROFIdrive sechs Anwendungsklassen, denen sich die meisten Antriebsanwendungen zuordnen lassen.

Tabelle 3- 1 Tabelle 3-1 Applikations-/Anwendungsklassen

Klasse	Antrieb
Klasse 1	Standardantriebe (wie z.B. Pumpen, Lüfter, Rührwerke etc.); realisiert in SIMOTION und SINAMICS
Klasse 2	Standardantriebe mit Technologiefunktionen
Klasse 3	Positionierantriebe; realisiert in SIMOTION und SINAMICS
Klasse 4	Motion Control Antriebe mit zentraler, übergeordneter Motion Control Intelligenz; realisiert in SIMOTION und SINAMICS
Klasse 5	Motion Control Antriebe mit zentraler, übergeordneter Motion Control Intelligenz und patentiertem Lagerregelkonzept "Dynamic Servo Control" (DSC) ; realisiert in SIMOTION und SINAMICS
Klasse 6	Motion Control Antriebe mit dezentraler, in den Antrieben selber integrierter Motion Control Intelligenz

PROFIdrive definiert ein Gerätemodell aus Funktionsmodulen, die geräteintern zusammenarbeiten und die Intelligenz des Antriebssystems widerspiegeln.

Diesen Modulen sind Objekte zugeordnet, die im Profil beschrieben und hinsichtlich ihrer Funktionen definiert werden. Die gesamte Funktionalität eines Antriebs ist somit durch die Summe seiner Parameter beschrieben.

Im Gegensatz zu anderen Antriebsprofilen definiert PROFIdrive nur die Zugriffsmechanismen auf die Parameter sowie einen Subset von ca. 70 Profilparametern, wozu unter anderen z.B. Störpuffer, Antriebssteuerung und Geräteidentifikation gehören.

Alle anderen Parameter sind herstellerspezifisch, was den Antriebsherstellern große Flexibilität bei der Realisierung der Regelungsfunktionen gibt. Der Zugriff auf die Elemente eines Parameters erfolgt azyklisch über den sogenannten Base Mode Parameter Access.

Hinweis

Die Aufträge mit Base Mode Parameter Access sind so kodiert, wie Sie es vielleicht von Datensatz 47 (DPV1-Kommunikation von PROFIBUS) her kennen. Für eventuelle Unterschiede, siehe Spezifika bei PROFIBUS und PROFINET IO (Seite 44).

PROFIdrive für PROFIBUS nutzt als Kommunikationsprotokoll DP-V0, DP-V1 und die DP-V2-Erweiterungen für PROFIBUS mit den darin enthaltenen Funktionen Slave-Querverkehr und Taktsynchronisation.

PROFIdrive für PROFINET enthält die Funtionen IO-Controller - IO-Device Kommunikation und Taktsynchronisation.

Anwendungsklassen

Für hochdynamische und hochkomplexe Motion Control Aufgaben ist die Anwendungsklasse 4 die wichtigste. Diese Anwendungsklasse beschreibt im Detail die Master-Slave-Beziehung zwischen dem Controller und den Antrieben, die über PROFIBUS oder PROFINET miteinander verbunden sind.

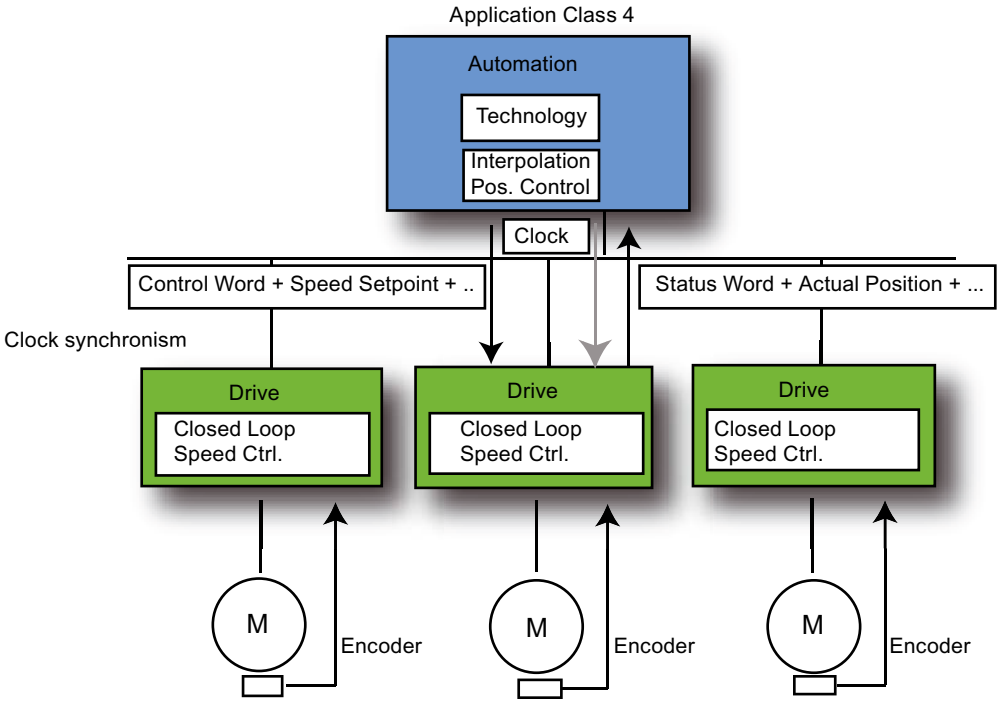


Bild 3-3 Anwendungsklassen

Durch die Funktion DSC (Dynamic Servo Control) wird die Dynamik und die Steifigkeit des Lageregelkreises deutlich verbessert. Bei SIMOTION normalerweise um die bei Drehzahl-Sollwert-Schnittstellen auftretenden Totzeiten (Sendezeit, Rechenzeit für Controller und Device), die durch ein zusätzliches, relativ einfaches Rückkoppelnetzwerk im Antrieb minimiert werden. Der Lageregler wird dabei im Antrieb vom SIMOTION Controller über eine Vorsteuerung und einer Positionsabweichung vorgesteuert, was sehr schnelle Lagereglertakte (z.B. bei Servo 125 µs im SINAMICS S) ermöglicht, und Totzeiten ausschließlich auf das Führungsverhalten reduziert.

3.5 PROFIdrive-spezifische Datentypen

Beschreibung

Zur Verwendung der PROFIdrive-konformen Kommunikation sind eine Reihe von Datentypen definiert. Detaillierte Informationen dazu finden Sie in folgenden Normen:

- IEC 61800-7-203
- IEC 61800-7-303
- IEC 61158-5

In diesen Normen finden Sie die Datentypen ausführlich definiert. Nachfolgend werden die wichtigsten Datentypen aufgelistet. Die Datentypen werden z.B. von der Funktion _readDriveParameterDescription geliefert. Weitere Informationen finden Sie auch unter

Hinweis

Für die S7 Kommunikation bzw. die Kommunikation mit SINAMICS müssen Sie mit der Systemfunktion *AnyType_to_BigByteArray* bzw. *BigByteArray_to_AnyType* eine Typkonvertierung bei verschiedenen Datentypen (Normalisierter Wert N2, N4; Normalisierter Wert X2, X4; Festkommawert E2 und Zeitkonstanten T2 und T4) durchführen.

PROFIdrive-Profil-spezifische Datentypen

Im PROFIdrive Profil verwendete Datentypen	Definition	Codierung (dez)
Boolean	Boolean (IEC 61158-5)	1
Integer8	Integer8 (IEC 61158-5)	2
Integer16	Integer16 (IEC 61158-5)	3
Integer32	Integer32 (IEC 61158-5)	4
Unsigned8	Unsigned8 (IEC 61158-5)	5
Unsigned16	Unsigned16 (IEC 61158-5)	6
Unsigned32	Unsigned32 (IEC 61158-5)	7
FloatingPoint32	Float32 (IEC 61158-5)	8
FloatingPoint64	Float64 (IEC 61158-5)	15
VisibleString	VisibleString (IEC 61158-5)	9
OctetString	OctetString (IEC 61158-5)	10

Im PROFIdrive Profil verwendete Datentypen	Definition	Codierung (dez)
TimeOfDay (mit Datumsanzeige)	TimeOfDay (IEC 61158-5)	11
TimeDifference	TimeDifference (IEC 61158-5)	12
Date	Date (IEC 61158-5)	13
TimeOfDay (without data indication)	TimeOfDay (IEC 61158-5)	52
TimeDifference (with data indication)	TimeDifference (IEC 61158-5)	53
TimeDifference (without data indication)	TimeDifference (IEC 61158-5)	54
Spezifische Datentypen	**Beschreibung siehe unten**	
N2 (Normalisierter Wert (16 Bit)		113
N4 (Normalisierter Wert (32 Bit)		114
V2 Bit Sequenz		115
L2 Nibble		116
R2 Rezipoke Zeitkonstante		117
T2 Zeitkonstante (16 Bit)		118
T4 Zeitkonstante (32 Bit)		119
D2 Zeitkonstante		120
E2 Festkommawert (16 Bit)		121
C4 Festkommawert (32 Bit)		122
X2 Normalisierter Wert, variabel (16 Bit)		123
X4 Normalisierter Wert, variabel (32 Bit)		124

Normalisierter Wert N2, N4

Linear normalisierter Wert, 0% korrespondiert mit 0 (0x0), 100% korrespondiert mit 2^{12} (0x4000) für N2, oder 2^{28} (0x40000000) für N4. Die Länge beträgt 2 bzw. 4 Oktetts.

Codierung

Darstellung in 2er Komplementen, MSB (Most Signifikant Bit) ist das erste Bit nach dem Vorzeichen-Bit (SN) des ersten Oktetts.

- SN = 0; positive Zahlen mit 0
- SN = 1; negative Zahlen

Wertebereich N2, N4	Auflösung N2, N4	Cod. N2, N4 (Dez)	Oktett	Bit							
				8	7	6	5	4	3	2	1
-200% ≤ i ≤ (200-2^{-14})%	2^{-12} = 0,0061 %	113	1	SN	2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
			2	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}
-200% ≤ i ≤ (200-2^{30})%	2^{-28} = 9,3 * 10^{-8}%	114	3	2^{-15}	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}
			4	2^{-23}	2^{-24}	2^{-25}	2^{-26}	2^{-27}	2^{-28}	2^{-29}	2^{-30}

Normalisierter Wert X2, X4 (Beispiel X=12/28)

Linear normalisierter Wert, 0% korrespondiert mit 0 (0x0), 100% korrespondiert mit 2^x. Diese Strukturen sind identisch zu N2 und N4, nur dass die Normalisierung variabel ist. Die Normalisierung kann aus den Parameterbeschreibungen ermittelt werden. Die Länge beträgt 2 bzw. 4 Oktetts.

Codierung

Darstellung in 2er Komplementen, MSB (Most Significant Bit) ist das erste Bit nach dem Vorzeichen-Bit (SN) des ersten Oktetts.

- SN = 0; positive Zahlen mit 0
- SN = 1; negative Zahlen

Wertebereich X2, X4	Auflösung X2, X4	Cod. X2, X4 (Dez)	Oktett	Bit							
				8	7	6	5	4	3	2	1
-800% ≤ i ≤ 800-2^{-12})%	2^{-12}	123	1	SN	2^{2}	2^{1}	2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}
			2	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}
-800% ≤ i ≤ 800-2^{-28})%	2^{-28}	124	3	2^{-13}	2^{-14}	2^{-15}	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}
			4	2^{-21}	2^{-22}	2^{-23}	2^{-24}	2^{-25}	2^{-26}	2^{-27}	2^{-28}

Festkommawert E2

Linearer Festkommawert mit vier Stellen nach dem Dezimalpunkt. 0 korrespondiert mit 0 (0x0), 128 korrespondiert mit 2^{14} (0x4000). Die Länge beträgt 2 Oktetts.

Codierung

Darstellung in 2er Komplementen, MSB (Most Signifikant Bit) ist das erste Bit nach dem Vorzeichen-Bit (SN) des ersten Oktetts.

- SN = 0; positive Zahlen mit 0
- SN = 1; negative Zahlen

Wertebereich E2	Auflösung	Cod (dez)	Oktett	Bit							
				8	7	6	5	4	3	2	1
-256+2^{-7} ≤ i ≤ 256-2^{-7}	2^{-7} = 0,0078125	121	1	SN	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}
			2	2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}

Festkommawert C4

Linearer Festkommawert mit vier Stellen nach dem Dezimalpunkt. 0 korrespondiert mit 0 (0x0), 0,0001 korrespondiert mit 2^{0} (0x0000 0001).

Codierung

Wie bei Integer32, die Wichtung der Bits wurde um den Faktor 10000 reduziert.

Wertebereich	Auflösung	Codierung (dez)	Länge
-214748,3648 ≤ i ≤ 214748,3648	10^{-4} = 00001	122	4 Oktett

Bitsequenz V2

Bitsequenz zur Kontrolle und Darstellung von Applikationsfunktionen. 16 Boolsche Variablen sind zu 2 Oktetts kombiniert.

Wertebereich	Auflösung	Cod (dez)	Oktett	Bit							
				8	7	6	5	4	3	2	1
		115	1	15	14	13	12	11	10	9	8
			2	7	6	5	4	3	2	1	0

Nibble (Halbbyte) L2

Vier zusammengehörige Bits ergeben ein Nibble. Vier Nibbles sind durch zwei Oktetts repräsentiert.

Codierung

Wertebereich	Auflösung	Cod (dez)	Oktett	Bit							
				8	7	6	5	4	3	2	1
-	-	116	1	Nibble 3				Nibble 2			
			2	Nibble 1				Nibble 0			

Zeitkonstanten T2 und T4

Zeitdaten als Vielfaches der Abtastzeit T_a. Interpretierter Wert = Interner Wert * T_a

Codierung

- T2: Wie bei Unsigned16 mit eingeschränktem Wertebereich $0 \leq x \leq 32767$
  Wenn interpretiert, werden interne Werte, die außerhalb des Wertebereichs liegen, auf 0 gesetzt.
- T4: Wie bei Unsigned32

Die Werte von Zeitparametern der Typen D2, T2, T4 und R2 beziehen sich immer auf die spezifizierte konstante Abtastzeit T_a. Die zugehörige Abtastzeit (Parameter p0962) wird für die Interpretation des internen Wertes benötigt.

Wertebereich	Auflösung	Codierung (dez)	Länge
$0 \leq i \leq 32767 * T_a$	T_a	118	2 Oktett
$0 \leq i \leq 4294967295 * T_a$	T_a	119	4 Oktett

Zeitkonstante D2

Zeitdaten als Bruchteil der konstanten Abtastzeit T_a. Interpretierter Wert = Interner Wert * T_a/16348

Codierung

- T2: Wie bei Unsigned16 mit eingeschränktem Wertebereich 0 ≤ x ≤ 32767
  Wenn interpretiert, werden interne Werte, die außerhalb des Wertebereichs liegen, auf 0 gesetzt.

Wertebereich	Auflösung	Codierung (dez)	Länge
0 ≤ i ≤ (2-2-14) * T_a	T_a	120	2 Oktett

Zeitkonstante R2

Zeitdaten als reziprokes Vielfaches der konstanten Abtastzeit T_a. Interpretierter Wert = 16348 * T_a/ Interner Wert

Codierung

- T2: Wie bei Unsigned16 mit eingeschränktem Wertebereich 0 ≤ x ≤ 16384
  Wenn interpretiert, werden interne Werte, die außerhalb des Wertebereichs liegen, auf 16384 gesetzt.

Wertebereich	Auflösung	Codierung (dez)	Länge
1 * Ta ≤ i ≤ 16384 * T_a	T_a	117	2 Oktett

Siehe auch

Parameter-Request-/Response Datensatz (Seite 40)

3.6 Azyklische Kommunikation (Base Mode Parameter Access)

3.6.1 Azyklische Kommunikation

Beschreibung

PROFIdrive-Antriebsgeräte werden mit Steuer-Signalen und Soll-Werten von der Steuerung versorgt und liefern Status-Signale und Ist-Werte.

Diese Signale werden normalerweise zyklisch (d.h. ständig) zwischen Steuerung und Antrieb übertragen.

Daneben kennen PROFIdrive Antriebsgeräte Parameter, in denen weitere benötigte Daten gehalten werden, etwa Fehler-Codes, Warnungen, Reglerparameter, Motordaten,... Diese Daten werden normalerweise nicht zyklisch (d.h. ständig) übertragen, sondern nur bei Bedarf "azyklisch". Auch Befehle an den Antrieb können durch Parameterzugriffe übermittelt werden.

Grundsätzlich erfolgt das Schreiben/Lesen von Parametern aus PROFIdrive-Antrieben azyklisch über den sogenannten "Base Mode Parameter Access". Der "Base Mode Parameter Access" kann sowohl mit PROFIBUS als auch mit PROFINET benutzt werden. Für die Unterschiede zwischen PROFIBUS und PROFINET, siehe Spezifika bei PROFIBUS und PROFINET IO (Seite 44).

Dieser Dienst wird von PROFIdrive definiert und bereitgestellt und kann parallel zu der zyklischen Kommunikation auf dem jeweiligen Bus zum Einsatz kommen. Das PROFIdrive Profil legt fest, wie genau dieser grundlegenden Mechanism genutzt wird für den Schreib-Lesezugriff auf Parameter eines PROFIdrive-konformen Antriebs.

3.6.2 Parameter lesen und schreiben mit Base Mode Parameter Access

Beschreibung

Die Kommunikation zum Schreiben/Lesen von Parametern erfolgt bei PROFIdrive-Antrieben, wie z.B. SINAMICS S120, demnach immer über den Base Mode Parameter Access, dessen Aufbau im PROFIdrive Profil definiert ist. Der Aufbau findet sich z.B. auch im SINAMICS S120 Funktionshandbuch unter **Azyklische Kommunikation**.

Ein Parameterzugriff besteht dabei immer aus einem Pärchen:

- Write Request ("Datensatz Schreiben")
- Read Request ("Datensatz Lesen")

Die Reihenfolge muss eingehalten werden, egal ob lesender oder schreibender Zugriff.

Mit einem "Datensatz Schreiben" wird der Parameterauftrag übermittelt (z.B. einen Parameter x lesen). Mit einem "Datensatz Lesen", wird die Antwort auf diesen Parameterauftrag abgeholt (Wert des Parameters x).

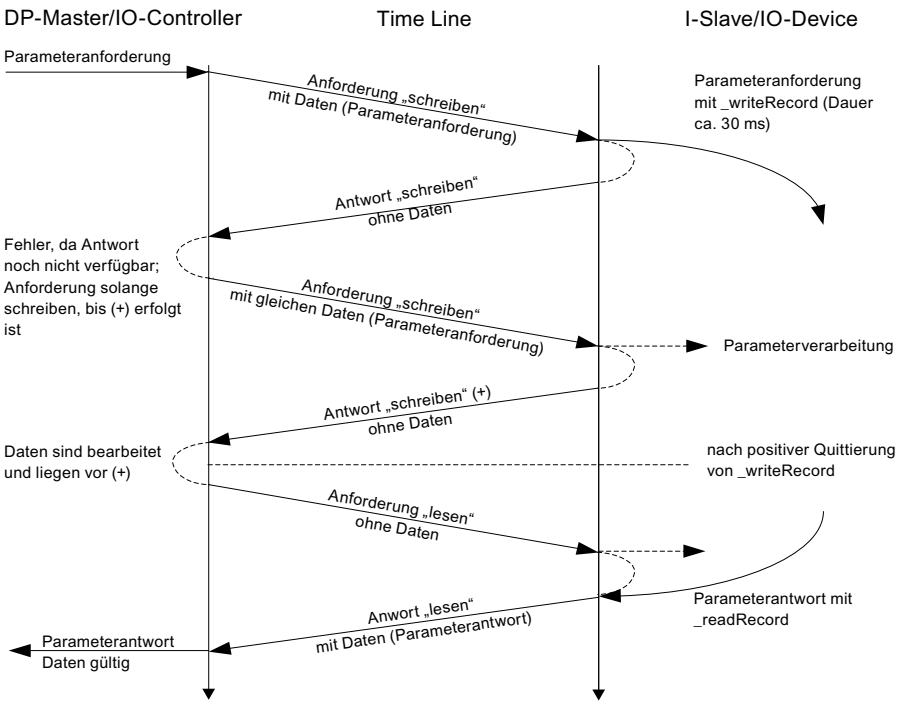


Bild 3-4 Azyklisch lesen und schreiben

Aus der Abbildung **Azyklisch lesen und schreiben** geht hervor, dass sowohl "Datensatz Schreiben" als auch "Datensatz Lesen" jeweils aus folgenden Teilen besteht:

- Request ("Anforderung")
- Response ("Antwort")

Die Steuerung wertet diese "Request Reference" nicht aus. Das Anwenderprogramm kann bzw. sollte diese Referenz jedoch auswerten.

Schreiben von Parametersätzen

Zum Schreiben von Parameterwerten (einer oder mehrere) werden zunächst der Auftragsstruktur Daten (P-Request/Response Datensatz) übergeben, die dann per "Datensatz Schreiben" mit _writeRecord übertragen werden. Durch wiederholtes "Datensatz lesen" (_writeRecord ohne Daten) kann der Status überprüft werden, bis eine positive Quittierung kommt. Danach wird _readRecord ("Datensatz lesen") solange gesendet, bis der Slave die Daten liefert.

Hinweis

Ein "Datensatz Schreiben" ohne Daten dient dazu den Status von "Datensatz schreiben" mit Daten zu ermitteln, bis die positive Quitter

Eine erfolgreiche Beendigung von "Datensatz Schreiben" signalisiert nur die fehlerfreie Übertragung des Datensatzes über den Kommunikationsweg, aber nicht die fehlerfreie Ausführung des Vorgangs im Zielgerät.

Lesen von Parametersätzen

Zum Lesen von Parameterwerten wird zunächst der Datenblock zusammengestellt, welche(r) Parameter gelesen werden soll(en). Dieser Datensatz wird über das Paar "Datensatz Schreiben"-"Datensatz Lesen" (zuerst _writeRecord und dann _readRecord) an den Antrieb übertragen. Ein nachfolgendes "Datensatz Lesen" liefert dann **einmalig** die angeforderten Werte (die gleiche Auftragsreferenz wird in der Antwort auch mitgeliefert).

Die Abläufe finden Sie oben auch bildlich dargestellt.

Im PROFIdrive Profil wird festgelegt, wie Daten übertragen werden, die größer als 1 Byte sind. Es gilt das sog. "Big Endian"-Format, das die höchstwertigen Teile zuerst überträgt:

WORD		High Byte (Byte 1)	Low Byte (Byte 2)
DOUBLE WORD	High Word	High Byte (Byte 1)	Low Byte (Byte 2)
	Low Word	High Byte (Byte 3)	Low Byte (Byte 4)

Darstellung WORD und DWORD im Big Endian Format

Da die Steuerung ggf. andere interne Datenrepräsentationen hat, ist es dann erforderlich, dass bei der Zusammenstellung und Auswertung der Daten im P-Request-/Response Datenblock (Datensatz 47) eine Konvertierung explizit vorgenommen wird.

Bei SIMOTION ist eventuell eine Konvertierung erforderlich, siehe Programmbeispiel.

Siehe auch

Regel 5 - Maximal acht Aufrufe gleichzeitig in SIMOTION möglich (Seite 57)

Parameter-Request-/Response Datensatz (Seite 40)

3.6.3 Parameter-Request-/Response Datensatz

Aufbau des P- Request -/ Response Datensatzes

Der besteht grundsätzlich aus:

- Einem Header (Auftragskennung, Ziel-Achse/Drive-Objekt, Anzahl Parameter im Auftrag)
- Einer Request Reference; Auftragsreferenz, um einen Auftrag zu identifizieren
- Auftragsdaten (Attribut, Anzahl Elemente/Indices, Parameternummer und -subindex), bei Schreibaufträgen auch die Werte.
- Werten, die der Funkion übergeben werden

Für einen WRITE- bzw. READ-Request haben die übertragenen Daten folgenden Aufbau.

	Parameter Auftrag	Byte n+1	Byte	Offset
Header	Request-Header	Request Reference	RequestID	0
		Achse	Anzahl Parameter	2
Details	1. Parameter	Attribut	Anzahl Elemente	4
		Parameternummer		6

	Parameter Auftrag	Byte n+1	Byte	Offset
		Subindex		8
	...			
	n. Parameter	Attribut	Anzahl Elemente	
		Parameternummer		
		Subindex		
Werte nur beim Schreiben	1. Parameterwert(e)	Format	Anzahl Werte	
		Werte		
	...			
	n. Parameterwert(e)	Format	Anzahl Werte	
		Werte		
		...		

Aufbau nach Base Mode Parameter Access - Parameter Request

Für den anschließenden Parameter Response ist folgender Aufbau festgelegt. Dieser muss mit _readRecord abgeholt werden.

	Parameter Antwort	Byte n+1	Byte	Offset
	Request-Header	Request Reference gespiegelt	RequestID gespiegelt oder Fehler	0
		Achs-Nr / DO-ID gespiegelt	Anzahl Parameter	2
Werte nur beim Lesen	1. Parameterwert(e)	Format	Anzahl Werte	4
		Werte oder Fehlercodes		6
		...		
	...			
Fehlerwerte nur bei negativer Antwort	n. Parameterwert(e)	Format	Anzahl Werte	
		Werte oder Fehlercodes		
		...		

Aufbau nach Base Mode Parameter Access - Parameter Response

Die genaue Codierung der einzelnen Bestandteile der Datenstruktur kann dem PROFIdrive Profil oder dem Funktionshandbuch SINAMICS S120 entnommen werden. Wichtig ist die Zuordnung von "Request" und "Response" sowie "Datensatz Schreiben" und "Datensatz Lesen" über die Auftragsreferenz "Request Reference" in obigerTabelle.

Request Reference

Die "Request Reference" wird für die Zuordnung von Write Request zu nachfolgendem Read Request, da prinzipiell die Steuerung mehrere Vorgänge parallel (bis zu 8) zu unterschiedlichen Zielgeräten über denselben Feldbus abwickeln kann.

Beschreibung der Felder Parameterauftrag und -antwort

Feld	Datentyp	Werte	Bemerkung
Auftragsreferenz	Unsigned8	0x01 ... 0xFF	
	Eindeutige Identifizierung des Auftrag-/Antwortpaares für den Master. Der Master ändert die Auftragsreferenz mit jedem neuen Auftrag. Der Slave spiegelt die Auftragsreferenz in seiner Antwort.		
Auftragskennung	Unsigned8	0x01 0x02	Leseauftrag Schreibauftrag
	Gibt an, um welchen Auftrag es sich handelt. Beim Schreibauftrag werden die Änderungen im flüchtigen Speicher (RAM) ausgeführt. Zur Übernahme der geänderten Daten in den nichtflüchtigen Speicher muss ein Speichervorgang ausgeführt werden (p0971, p0977).		
Antwortkennung	Unsigned8	0x01 0x02 0x81 0x82	Leseauftrag (+) Schreibauftrag (+) Leseauftrag (-) Schreibauftrag (-)
	Spiegelung der Auftragskennung mit der Zusatzinformation, ob der Auftrag positiv oder negativ ausgeführt wurde. Negativ bedeutet: Der Auftrag konnte ganz oder teilweise nicht ausgeführt werden. Es werden pro Teilantwort statt der Werte die Fehlerwerte übertragen.		
Antriebsobjekt–Nummer	Unsigned8	0x01 ... 0xFE	Nummer
	Vorgabe der Antriebsobjekt–Nummer bei einem Antriebsgerät mit mehreren Antriebsobjekten. Es kann über die gleiche DPV1-Verbindung auf verschiedene Antriebsobjekte mit je einem eigenen Parameternummernbereich zugegriffen werden.		
Anzahl Parameter	Unsigned8	0x01 ... 0x27	Anzahl 1 ... 39 Begrenzt durch DPV1-Telegrammlänge
	Definiert bei Multiparameterauftrag die Anzahl der folgenden Bereiche Parameteradresse und/oder Parameterwert. Für einfache Aufträge ist Anzahl Parameter = 1.		
Attribut	Unsigned8	0x10 0x20 0x30	Wert Beschreibung Text (Nicht implementiert bei SINAMICS)
	Art des Parameterelements, auf das zugegriffen wird.		
Anzahl Elemente	Unsigned8	0x00 0x01 ... 0x75	Sonderfunktion Anzahl 1 ... 117 Begrenzt durch DPV1-Telegrammlänge
	Anzahl der Arrayelemente, auf die zugegriffen wird.		
Parameternummer	Unsigned16	0x0001 ... 0xFFFF	Nummer 1 ... 65535
	Adressiert den Parameter, auf den zugegriffen wird.		
Subindex	Unsigned16	0x0000 ... 0xFFFE	Nummer 0 ... 65534
	Adressiert das erste Arrayelement des Parameters, auf das zugegriffen wird.		

Feld	Datentyp	Werte	Bemerkung
Format	Unsigned8	0x02 0x03 0x04 0x05 0x06 0x07 0x08 Andere Werte 0x40 0x41 0x42 0x43 0x44	Datentyp Integer8 Datentyp Integer16 Datentyp Integer32 Datentyp Unsigned8 Datentyp Unsigned16 Datentyp Unsigned32 Datentyp FloatingPoint Siehe PROFIdrive Profile V3.1 Zero (ohne Werte als positive Teilantwort eines Schreibauftrags) Byte Word Double word Error
	Format und Anzahl spezifizieren den nachfolgend durch Werte belegten Platz im Telegramm. Beim Schreibvorgang sind bevorzugt Datentypen nach PROFIdrive Profile anzugeben. Ersatzweise sind auch Byte, Wort und Doppelwort möglich.		
Anzahl Werte	Unsigned8	0x00 ... 0xEA	Anzahl 0 ... 234 Begrenzt durch DPV1-Telegrammlänge
	Gibt die Anzahl der folgenden Werte an.		
Fehlerwerte	Unsigned16	0x0000 ... 0x00FF	Bedeutung der Fehlerwerte --> siehe Tabelle 4-29
	Die Fehlerwerte bei negativer Antwort. Wenn die Werte aus einer ungeraden Anzahl von Bytes bestehen, wird ein Nullbyte angehängt. Damit wird die Wortstruktur des Telegramms sichergestellt.		
Werte	Unsigned16	0x0000 ... 0x00FF	
	Die Werte des Parameters beim Lesen oder Schreiben. Wenn die Werte aus einer ungeraden Anzahl von Bytes bestehen, wird ein Nullbyte angehängt. Damit wird die Wortstruktur des Telegramms sichergestellt.		

Für weitere Informationen zur Codierung von PROFIdrive Datentypen, siehe PROFIdrive-spezifische Datentypen (Seite 33)

Siehe auch

Programmbeispiel (Seite 71)

Parameter lesen und schreiben mit Base Mode Parameter Access (Seite 38)

3.6.4 Spezifika bei PROFIBUS und PROFINET IO

Globale und lokale Parameter

Nach PROFIdrive werden zwei Parameterbereiche unterschieden:

- Globale Parameter; diese sind dem ganzen Antriebsgerät zugeordnet. Wenn Sie verschiedene DOs eines Antriebsgerätes adressieren möchten, zeigt ein globaler Parameter immer denselben Wert.
- Lokale Parameter; diese Parameter sind achs- bzw. DO-spezifisch. Die achs- und DO-spezifischen Parameter können für jedes Achs-DO verschiedene Werte haben.

Entsprechend leiten sich die beiden Zugriffsarten für den Base Mode Parameter Access ab:

- Base Mode Parameter Access - Local
- Base Mode Parameter Access - Global

Spezifische Eigenschaften einer azyklischen Kommunkation bei PROFIBUS

Für Kommunikation über PROFIBUS wird der Datensatz 47 (0x002F) zum Zugriff auf Parameter in PROFIdrive Antrieben verwendet.

- Base Mode Parameterr Access – Global; die Parameter (aller DOs, globale und lokale Parameter) des Antriebsgerätes sind über die PAPs des Antriebsgerätes ansprechbar.

Hinweis

Die PROFIdrive Norm legt fest, dass in PROFIdrive Antrieben kein Pipelining von Aufträgen unterstützt wird, dass also daher zu einem Antriebsgerät immer nur ein "Datensatz Schreiben/Lesen" gleichzeitig möglich ist. Sind aber mehrere PROFIdrive-Antriebsgeräte an einer Steuerung über PROFIBUS angeschlossen, so kann zu jedem dieser Antriebsgeräte je ein Auftrag zeitlich parallel abgewickelt werden. Die Höchstanzahl ist dann steuerungsabhängig. Daten für SIMOTION sind in Regel 5 - Maximal acht Aufrufe gleichzeitig in SIMOTION möglich (Seite 57) angegeben.

Spezifische Eigenschaften einer azyklischen Kommunikation bei PROFINET IO

Bei Verwendung von PROFINET ändert sich an den grundlegenden Abläufen nichts, jedoch lautet die Datensatz-Nummer dann 0xB02E ("Base Mode Parameter Access - Local") und 0XB02F("Base Mode Parameter Access - Global").

- Base Mode Parameter Access - Local; Für einen Zugriff auf globale Parameter eines DOs können Sie jede gültige PAP des Antriebsgerätes benutzen. Für einen lokalen Zugriff wird die lokale DO-ID (Achs-ID) nicht ausgewertet. Diese können Sie dann über die Diagnoseadresse des PAP adressieren.
- Base Mode Parameter Access - Global; Für einen erfolgreichen Zugriff auf die lokalen Parameter eines DOs benötigen Sie eine gültige DO-ID. Für den Zugriff auf die globalen Parameter eines Antriebsgerätes können Sie jede gültige PAP benutzen.

3.6.5 Fehlerauswertung

Beschreibung

Es können zwei unterschiedliche Arten von Fehlern im Zusammenhang mit den Base Mode Parameter Access Diensten auftreten:

- Fehler in der Kommunikation (Übermittlung der Daten)

  Beispielsweise könnte das adressierte Gerät nicht vorhanden oder eingeschaltet sein. Diese Art von Fehlern wird über Rückgabewerte der Systemfunktionen übermittelt und kann der Beschreibung der Systemfunktionen in den SIMOTION Referenzlisten entnommen werden.

- Fehler bei der Bearbeitung der Aufträge selber

  Beispielsweise könnte versucht werden, auf einen Read-Only-Parameter schreibend zuzugreifen.

  Fehlercodes zu dieser zweiten Art von Fehlern sind für PROFIdrive-konforme Antriebe in der PROFIdrive Norm festgelegt und sind unten aufgelistet.

  Durch die Response ID 0x81 (hex) bzw. 0x82 (hex) wird ein Fehler beim Parameterzugriff gekennzeichnet.

  Die Fehlercodes werden in der Antwort des Antriebsgerätes im P-Response-/Request Datenblock zurückgeliefert, siehe Tabelle unten. Die Unterscheidung, ob es sich um einen Fehler-Code oder um einen "echten" Wert eines angefragten Parameters handelt, kann man in der Parameter Response dem Feld "Format" entnehmen, siehe Tabelle Aufbau nach Base Mode Parameter Access - Parameter Response (Seite 40), Offset 4, "Format".

  Die Codierung für das Feld "Format" können Sie ebenfalls dort entnehmen. Code 0x44 (hex) deutet auf einen Fehler-Code im Feld "Werte" hin. Andere Werte von "Format" legen fest, in welchem Zahlenformat (z.B. Bool, Byte, Integer8, ...) der Wert im Feld "Werte" zurückgeliefert wurde.

Hinweis

Die Fehlercodes bis 0x19 entsprechen dem PROFIdrive Profil. Die Fehlercodes ab 0x65 sind herstellerspezifisch und können deshalb von Antrieb zu Antrieb unterschiedlich sein.

Fehlercodes in Base Modes Parameter Access Antworten

Fehler-code	Bedeutung	Bemerkung	Zusatz-Info
0x00	Unzulässige Parameternummer.	Zugriff auf nicht vorhandenen Parameter.	–
0x01	Parameterwert nicht änderbar.	Änderungszugriff auf einen nicht änderbaren Parameterwert.	Subindex
0x02	Untere oder obere Wertgrenze überschritten.	Änderungszugriff mit Wert außerhalb der Wertgrenzen.	Subindex
0x03	Fehlerhafter Subindex.	Zugriff auf nicht vorhandenen Subindex.	Subindex
0x04	Kein Array.	Zugriff mit Subindex auf nichtindizierten Parameter.	–

Fehler-code	Bedeutung	Bemerkung	Zusatz-Info
0x05	Falscher Datentyp.	Änderungszugriff mit Wert, der nicht zum Datentyp des Parameters passt.	–
0x06	Kein Setzen erlaubt (nur zurücksetzen).	Änderungszugriff mit Wert ungleich 0, wo dies nicht erlaubt ist.	Subindex
0x07	Beschreibungselement nicht änderbar.	Änderungszugriff auf nicht änderbares Beschreibungselement.	Subindex
0x09	Beschreibungsdaten nicht vorhanden.	Zugriff auf nicht vorhandene Beschreibung (Parameterwert ist vorhanden).	–
0x0B	Keine Bedienhoheit.	Änderungszugriff bei fehlender Bedienhoheit.	–
0x0F	Kein Textarray vorhanden	Zugriff auf nicht vorhandenes Textarray (Parameterwert ist vorhanden).	–
0x11	Auftrag wegen Betriebszustand nicht ausführbar.	Zugriff ist aus nicht näher spezifizierten temporären Gründen nicht möglich.	–
0x14	Wert unzulässig.	Änderungszugriff mit Wert, der zwar innerhalb der Grenzen liegt, aber aus anderen dauerhaften Gründen unzulässig ist (Parameter mit definierten Einzelwerten).	Subindex
0x15	Antwort zu lang.	Die Länge der aktuellen Antwort überschreitet die maximal übertragbare Länge.	–
0x16	Parameteradresse unzulässig.	Unzulässiger oder nicht unterstützter Wert für Attribut, Anzahl Elemente, Parameternummer oder Subindex oder eine Kombination.	–
0x17	Format unzulässig.	Schreibauftrag: Unzulässiges oder nicht unterstütztes Format der Parameterdaten.	–
0x18	Anzahl Werte nicht konsistent.	Schreibauftrag: Anzahl Werte der Parameterdaten passen nicht mit Anzahl Elemente in der Parameteradresse zusammen.	–
0x19	Antriebsobjekt existiert nicht.	Zugriff auf ein Antriebsobjekt das nicht existiert.	–
0x65	Parameter momentan deaktiviert.	Zugriff auf einen Parameter, der zwar vorhanden ist, aber zum Zeitpunkt des Zugriffes keine Funktion erfüllt (z. B. n-Regelung eingestellt und Zugriff auf Parameter von U/f–Steuerung).	–
0x6B	Parameter %s [%s]: Kein Schreibzugriff bei freigegebenem Regler.	–	–
0x6C	Parameter %s [%s]: Unbekannte Einheit.	–	–
0x6D	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Geber (p0010 = 4).	–	–
0x6E	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Motor (p0010 = 3).	–	–
0x6F	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Leistungsteil (p0010 = 2).	–	–
0x70	Parameter %s [%s]: Schreibzugriff nur in Schnellinbetriebnahme (p0010 = 1).	–	–

Fehler-code	Bedeutung	Bemerkung	Zusatz-Info
0x71	Parameter %s [%s]: Schreibzugriff nur in Bereit (p0010 = 0).	–	–
0x72	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Parameter-Reset (p0010 = 30).	–	–
0x73	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Safety (p0010 = 95).	–	–
0x74	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Tech. Applikation/Einheiten (p0010 = 5).	–	–
0x75	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand (p0010 ungleich 0).	–	–
0x76	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Download (p0010 = 29).	–	–
0x77	Parameter %s [%s] darf im Download nicht geschrieben werden.	–	–
0x78	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Antriebskonfiguration (Gerät: p0009 = 3).	–	–
0x79	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Festlegung Antriebstyp (Gerät: p0009 = 2).	–	–
0x7A	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Datensatzbasis–Konfiguration (Gerät: p0009 = 4).	–	–
0x7B	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Geräte-Konfiguration (Gerät: p0009 = 1).	–	–
0x7C	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Geräte-Download (Gerät: p0009 = 29).	–	–
0x7D	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Geräte-Parameter-Reset (Gerät: p0009 = 30).	–	–
0x7E	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Gerät bereit (Gerät: p0009 = 0).	–	–
0x7F	Parameter %s [%s]: Schreibzugriff nur in Inbetriebnahmezustand Gerät (Gerät: p0009 ungleich 0).	–	–

Fehler-code	Bedeutung	Bemerkung	Zusatz-Info
0x81	Parameter %s [%s] darf im Download nicht geschrieben werden.	–	–
0x82	Übernahme der Steuerungshoheit ist über BI: p0806 gesperrt.	–	–
0x83	Parameter %s [%s]: Gewünschte BICO-Verschaltung unmöglich.	BICO-Ausgang liefert nicht Float-Wert, der BICO–Eingang benötigt aber Float.	–
0x84	Parameter %s [%s]: Parameteränderung gesperrt (siehe p0300, p0400, p0922)	–	–
0x85	Parameter %s [%s]: Keine Zugriffsmethode definiert.	–	–
0xC8	Unterhalb aktuell gültiger Grenze.	Änderungsauftrag auf einen Wert, der zwar innerhalb der "absoluten" Grenzen liegt, der aber unterhalb der aktuell gültigen unteren Grenze liegt.	–
0xC9	Oberhalb aktuell gültiger Grenze.	Änderungsauftrag auf einen Wert, der zwar innerhalb der "absoluten" Grenzen liegt, der aber oberhalb der aktuell gültigen oberen Grenze liegt (z. B. vorgegeben durch die vorliegende Umrichterleistung).	–
0xCC	Schreibzugriff nicht erlaubt.	Schreibzugriff nicht erlaubt, da Zugriffsschlüssel nicht vorhanden.	–

3.6.6 Zusatzinformationen zu den Parametern eines PROFIdrive-Antriebs

Beschreibung

Aus einem PROFIdrive Antriebsgerät kann man nicht nur die Werte von Parametern lesen, sondern auch Beschreibungen zu den Parametern.

Die Unterscheidung wird bei der Übermittlung des "Parameter Request" im P-Response-/Request Datenblock im Feld "Attribute" vorgenommen:

Attribute = 0x10 (hex)	Wert
Attribute = 0x20 (hex)	Parameter-Beschreibung "Parameter Description"
Attribute 0 0x30 (hex)	Parameter-Text

Wird statt des Wertes zu einem Parameter dessen "Parameter Description" angefordert, enthält das Feld "Value" in der "Parameter Response" die Beschreibung (Datentyp, ggf. Anzahl der Indices des Parameters, ...).

Hinweis

In der Regel sind Parameter-Beschreibungen nur lesbar.

3.6.7 Systembefehle in SIMOTION

3.6.7.1 SIMOTION Systembefehle _writeRecord/_readRecord

Beschreibung

Ein "Datensatz Schreiben" kann in SIMOTION mit dem Systembefehl _writeRecord() erfolgen. Ein "Datensatz Lesen" kann mit dem Systembefehl _readRecord() erfolgen. Damit ist es also möglich, Parameter in einem PROFIdrive-Antrieb zu lesen, schreiben oder dessen Beschreibung auszulesen.

Die Beschreibung der Systemfunktionen und deren Eingangsparameter und Rückgabewerte kann in der SIMOTION Systemdokumentation gefunden werden:

- C2xx Referenzliste
- D4XX Referenzliste
- P350 Referenzliste

Diese Systembefehle _write/_readRecord sind universell verwendbar, nicht nur für PROFIdrive-Antriebe, sondern auch für z.B. intelligente Sensoren am PROFIBUS oder andere Peripherie-Baugruppen, die die sog. DP-V1-Dienste von PROFIBUS unterstützen.

Hinweis

Bei SIMATIC lauten die entsprechenden Systemfunktionen

SFB52 WR_REC Datensatz Schreiben

SFB53 RD_REC Datensatz Lesen

Um die SIMOTION-Systembefehle _write/_readRecord nutzen zu können, ist Folgendes notwendig:

- PROFIBUS DP: Ein Zugriff über logische I/O-Adresse wie auch Diagnose-Adresse ist möglich.
- PROFINET IO: Ein Zugriff ist nur über die Diagnose-Adresse eines Parameter Access Points (PAP) möglich.

Weiterhin ist die DO-ID nur für Datensatz 47 (0x002f) und Global Access (PROFINET 0xb02f) relevant. Bei Local Access (PROFINET IO 0xb02e) ist die Diagnoseadresse des zugehörigen PAP relevant, die DO-ID wird nicht ausgewertet.

So ist z.B. im Zusammenhang mit PROFIdrive-Antrieben die Telegramm-Anfangsadresse des zyklisch ausgetauschten PROFIdrive-Telegramms an das Gerät nötig.

Hat ein Antrieb auf einem Antriebsgerät mehrere Achsen (mit einer gemeinsam genutzten PROFIBUS-Anschaltung), so benötigt man zusätzlich zur Unterscheidung der Achsen im selben Gerät noch die Achsnummer "Axis-No." bzw. "DO-ID" im Datensatz 47. Beispiele für solche mehrachsigen Antriebe sind der SIMODRIVE 611universal bzw. SINAMICS S120. Zur Bestimmung der "DO-ID" bei SINAMICS S120, siehe SINAMICS S120-Inbetriebnahmehandbuch **Azyklische Kommunikation**.

Über "Axis-No." bzw. "DO-ID" = 0 kann auf die sogenannten "globalen Parameter" zugegriffen werden. Beispiele für solche "globalen Parameter" sind:

- P0918: PROFIBUS Adresse
- P0964: Device Identifikation (u.a. Hersteller, Version, Anzahl der Achsen)
- P0965: Profil Nummer (welche PROFIdrive Version ist implementiert)
- P0978: Liste der DO-Ids (welche "Axis-No." bzw. "DO-ID" wurden eingestellt)

3.6.7.2 SIMOTION Systembefehle _writeDrive.../_readDrive...

Beschreibung

Während die Systemfunktionen _readRecord und _writeRecord universell einsetzbar sind für alle Geräte an PROFIBUS, die die sog. DP-V1-Dienste "Datensatz Schreiben/Lesen" unterstützen, sind die folgenden Befehle speziell abgestimmt auf PROFIdrive-Antriebe laut PROFIdrive Profil:

- _read/writeDriveParameter (liest/schreibt einen ggf. indizierten Antriebsparameter)
- _read/writeDriveMultiParameter (liest/schreibt mehrere ggf. indizierte Antriebsparameter zu einem Antrieb bzw. Antriebsobjekt)
- _readDriveFaults (liest den aktuellen Störpuffereintrag eines Antriebs bzw. Antriebsobjektes)
- _readDriveParameterDescription (liest die Beschreibungsdaten eines Parameter aus dem Antrieb bzw. Antriebsobjekt)
- _readDriveMultiParameterDescription (liest die Beschreibungsdaten von mehreren Parameter aus dem Antrieb bzw. Antriebsobjekt)

Die Befehle erstellen intern den für die für die einzelnen Funktionen benötigten Datensatz 47 laut PROFIdrive Profil anhand der vom Anwender beim Aufruf der Systemfunktionen übergebenen Parameter und wickeln die Kommunikation zum PROFIdrive-Antrieb über "Datensatz Schreiben/Lesen" selbstständig ab.

Die Beschreibung der Befehle kann in der SIMOTION Systemdokumentation gefunden werden, siehe die Referenzlisten der jeweiligen Plattform.

Siehe auch

Gültigkeitsbereich für die Regeln (Seite 59)

3.6.7.3 Vergleich der Systembefehle

Beschreibung

In der folgenden Tabelle sind die wichtigsten Unterschiede zwischen den beiden Gruppen von Systembefehlen dargestellt:

Befehlsgruppe	Vorteil	Nachteil
_readRecord _writeRecord	• Allgemein verwendbar, nicht nur für DP-V1-Dienste zu Antrieben • Setzt nur die Kenntnis irgendeiner I/O-Adresse *auf dem Antriebsgerät* sowie der "DO-ID" bzw. "Axis-No" auf dem Antriebsgerät voraus	• Anwender muss Datensatz selber zusammenstellen • Anwender muss zwei Aufrufe programmieren für Parameterzugriffe in einem PROFIdrive-Antrieb • Anwender muss sich ggf. um die Konvertierung der Daten selber kümmern • "DO-ID" bzw. "Axis-No" muss bekannt sein
_readDrive... _writeDrive...	• Speziell angepasst an die typischen Kommunikationen mit PROFIdrive Antrieben • Anwender muss Aufbau des Datensatz 47 nicht kennen • Geringerer Programmieraufwand für den Anwender bei Kommunikation zu Antrieben	• Setzt Vorhandensein bzw. Kenntnis einer I/O-Adresse *des jeweiligen Antriebsobjekts* voraus • Eine I/O-Adresse für eine Antriebsobjekt gibt es nur bei zyklischer Kommunikation (mit PROFIBUS) zu dem Antriebsobjekt, möglicherweise also z.B. nicht für I/O-Erweiterungsmodule TB30, TMxx für ausschließliche Verwendung im Antrieb • Anwender muss sich um die Konvertierung der Daten selber kümmern

Eigenschaften der Systembefehle

Die Verwendung der Antriebs-spezifischen Systembefehle _write/_readDrive... bietet für Sie einerseits mehr Komfort als die Verwendung der allgemeinen Befehle _write/_readRecord, da Sie sich nicht um den Aufbau des Datensatz 47 kümmern müssen und nicht die aufeinanderfolgenden Aufrufe von _writeRecord und _readRecord jeweils in Schrittketten programmieren müssen. Da der Aufbau der übertragenen Datensätze dem System nicht bekannt ist wegen der allgemeinen Verwendbarkeit dieser Systemfunktionen, müssen Sie beim Senden und Empfangen ggf. die Konvertierung in die Darstellung laut PROFIdrive Profil selber vornehmen, siehe Programmbeispiel (Seite 71).

Andererseits ist aber die Verwendung der Befehle _write/_readDrive... beschränkt auf Fälle, bei denen es einen zyklischen Datenverkehr zum jeweiligen Antriebsobjekt gibt, da diese als Eingabe-Parameter benötigt wird. Demgegenüber kann mit _write/_readRecord auch dann noch auf Antriebsobjekte zugegriffen werden, wenn kein zyklischer Datenverkehr existiert (oder dessen I/O-Adresse nicht bekannt ist in der Applikation). Dies gelingt mit _write/_readRecord deshalb, weil für das Zusammenstellen des Datensatz 47 die explizite Kenntnis der "DO-ID" bzw. "Axis-No." ausreicht sowie die Kenntnis irgendeiner I/O-Adresse

auf dem Gerät. Dies kann z.B. dann von Vorteil sein, wenn einzelne Antriebsobjekte nur Antriebs-intern verwendet werden (d.h. ohne zyklischen Telegrammverkehr zur Steuerung) oder diese bei "generischer Programmierung" nicht allgemein bekannt ist.

Ab V4.1 können Sie auch mit den Befehlen _write/_readDrive...auf Antriebsobjekte zugreifen, wenn kein zyklischer Datenverkehr existiert, da Sie die "DO-ID" bzw. "Axis-No" als Parameter übergeben können.

3.6.7.4 Löschen von _readDrive- und _writeDrive-Aufträgen

Beschreibung

Wenn Sie nicht korrekte Lese- bzw. Schreibaufträge, die z.B. mit _readDriveParameter aufgerufen wurden, abbrechen bzw. löschen möchten, können Sie folgende Funktionen verwenden:

- _abortReadWriteRecordJobs, für die Funktionen _readRecord bzw. _writeRecord
- _abortAllReadWriteDriveParameterJobs, für die Funktionen
  - _readDrive(Multi)ParameterDescription
  - _readDrive(Multi)Parameter
  - _writeDrive(Multi)Parameter
  - _readDriveFaults

Sie können die Funktionen aufrufen, ohne dass Sie die CommandID kennen, auslesen.

3.6.8 Regeln für die Anwendung von _readRecord und _writeRecord

3.6.8.1 Regel 1 - Auftrag hat eigene Auftragsreferenz

Jeder Auftrag bekommt eine eigene Auftragsreferenz

Dies ist erforderlich, um unterschiedliche Aufträge zuordnen zu können. Die Auftragsreferenz kann wieder verwendet werden, wenn die Zuordnung auch anderweitig, z.B. durch zeitliche Abfolge, klar ist.

3.6.8.2 Regel 2 - Systemfunktionen bei asynchroner Programmierung

Beschreibung

R2: Sie müssen bei asynchroner Programmierung die Systemfunktion wiederholt mit gleichen Ids aufrufen, bis zur Beendigung der Funktion ("Langläufer"). In der Abbildung **Fehlerfreier Ablauf mit den Systemfunktionen _readRecord und _writeRecord** wird die fehlerfreie Verwendung der Systemfunktionen _writeRecord und _readRecord am Beispiel der Kommunikation zu SINAMICS S120 dargestellt.

Die Kommunikation zum Schreiben und Lesen von Parametern erfolgt zum SINAMICS S120 immer über den Datensatz 47, dessen Aufbau in der Dokumentation zum SINAMICS S120 beschrieben ist, siehe Inbetriebnahmehandbuch SINAMICS S120, Azyklische Kommunikation.

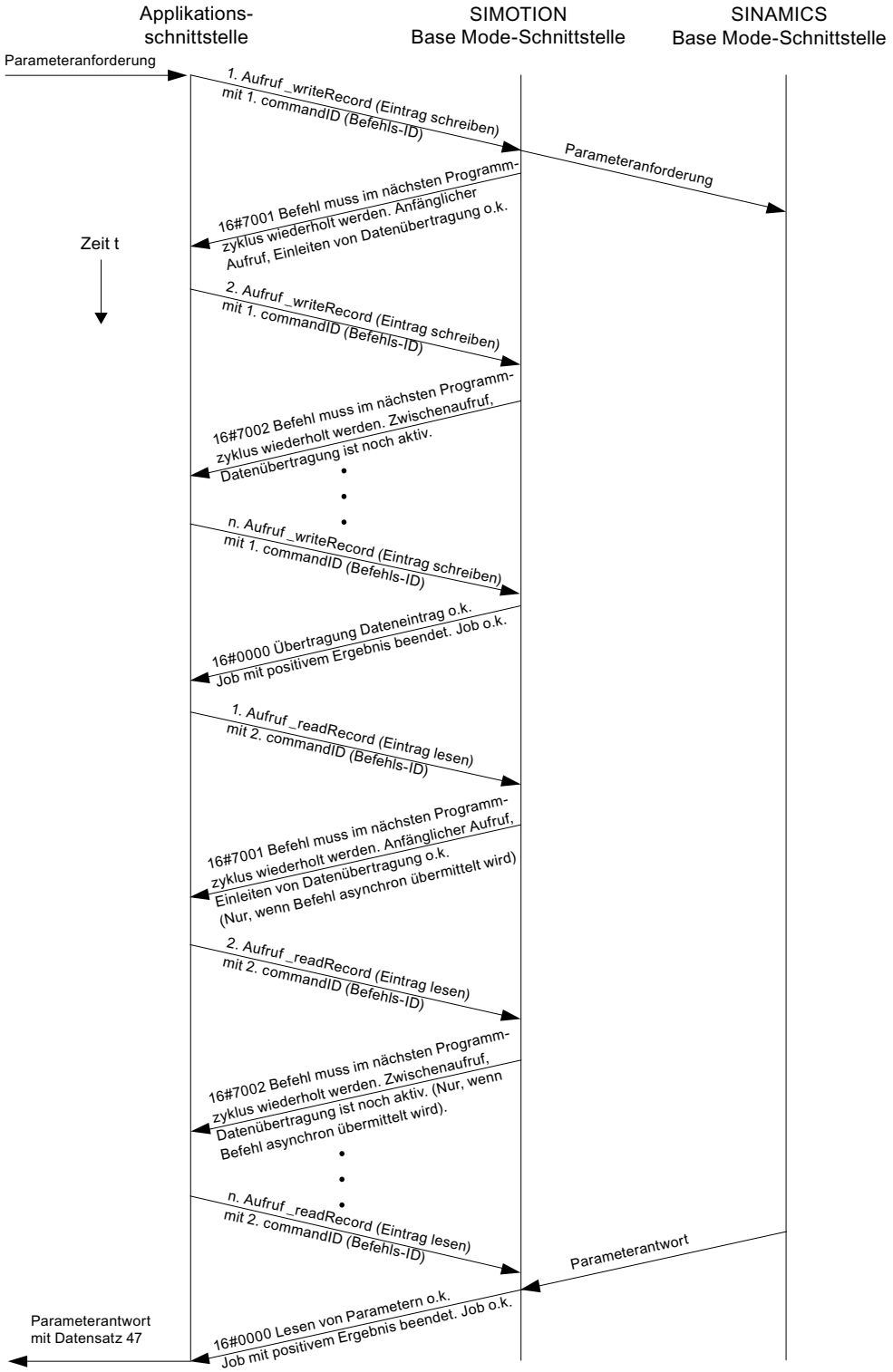


Bild 3-5 Fehlerfreier Ablauf mit den Systemfunktionen _readRecord und _writeRecord

3.6.8.3 Regel 3 - Datensatz Schreiben/Lesen pro PROFIDrive Antriebsgerät

Nur ein Datensatz Schreiben/Lesen pro PROFIdrive Antriebsgerät gleichzeitig

Im PROFIdrive Profil ist festgelegt, dass PROFIdrive-Antriebe kein Pipelining durchführen und daher nur ein Auftrag zu einer Zeit bearbeitet wird. Dies ist daher auch für SINAMICS S120 im Inbetriebnahmehandbuch so beschrieben.

Hinweis

Es ist dabei unerheblich, über welche Systemfunktionen die Übertragung in der Steuerung abgewickelt wird. Ein PROFIdrive-Antrieb kann nur einen Auftrag zu einer Zeit bearbeiten.

Hinweis

Es kann bei anderen Geräten am PROFIBUS durchaus möglich sein, dass diese mehrere "Datensatz Schreiben/Lesen" parallel unterstützen.

Hinweis

Da die Systemfunktionen _write/_readRecord universell einsetzbar sind, wird auf Steuerungsseite *nicht* verriegelt, dass nur ein "Datensatz Schreiben/Lesen" pro PROFIdrive-Antrieb zu einer Zeit angestoßen werden kann.

Folge für die Applikation auf der Steuerung:

Es muss verriegelt werden, dass die Applikation bzw. unterschiedliche Teile der Applikation gleichzeitig bzw. überlappend Aufträge an dasselbe PROFIdrive-Antriebsgerät senden, siehe auch Abschnitt unten Verriegelung von mehreren Aufrufen (Seite 60).

3.6.8.4 Regel 4 - Letzter Aufruf gewinnt bei SIMOTION

Bei SIMOTION "gewinnt" im Zweifel der letzte Aufruf

Wird die Regel 3 "Nur ein Datensatz Schreiben/Lesen pro PROFIdrive Antriebsgerät gleichzeitig" verletzt, indem zwischendurch ein zweiter _writeRecord Befehl auf denselben Antrieb abgesetzt wird, so kann die Antwort des ersten Auftrags anschließend nicht mehr gelesen werden. Der Versuch, die Antwort des Antriebs auf den ersten Auftrag zu lesen kann vom Antrieb nicht mehr bearbeitet werden und wird mit Fehler quittiert und abgebrochen. Der zeitliche Ablauf kann der Abbildung **Zweiter Aufruf von _writeRecord gewinnt im Zweifel** entnommen werden.

Um die Aufträge auf Steuerungsseite voneinander unterscheiden zu können, wurden für die Aufrufe der Systemfunktionen _writeRecord und _readRecord je getrennte commandID benutzt.

Um die Aufträge auch auf Antriebsseite voneinander unterscheiden zu können, wurde für den ersten und zweiten Auftrag eigene Auftragsreferenzen im Datensatz 47 vergeben.

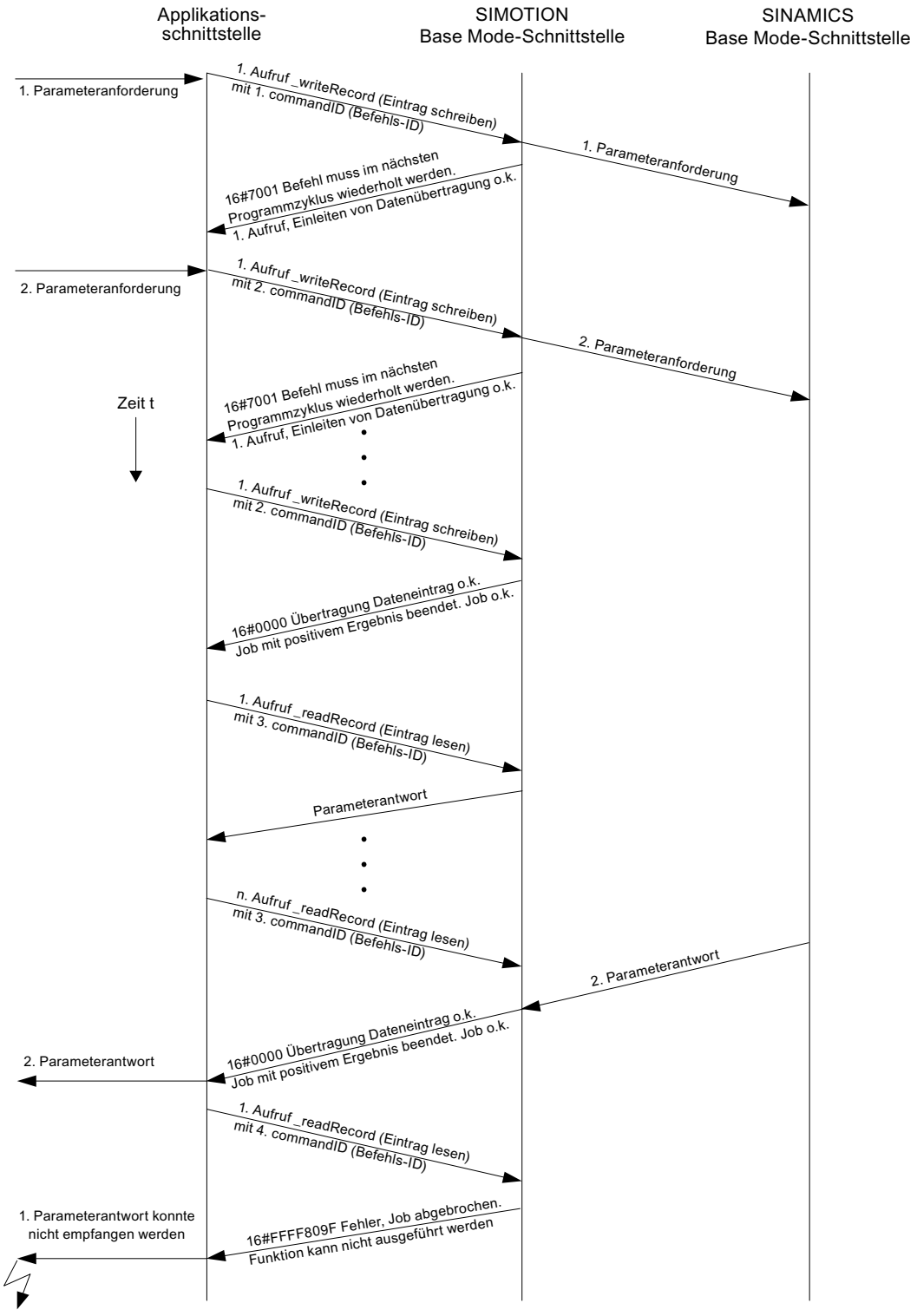


Bild 3-6 Zweiter Aufruf von _writeRecord gewinnt im Zweifel

3.6.8.5 Regel 5 - Maximal acht Aufrufe gleichzeitig in SIMOTION möglich

SIMOTION kann max. 8 Aufrufe von _write/_readRecord gleichzeitig verwalten

Obwohl laut Regel 3 (siehe Regel 3 (Seite 55)) nur ein Auftrag zu einer Zeit von *einem* PROFIdrive-Antriebsgerät gleichzeitig abgewickelt werden kann, ist es doch möglich, vom Steuerungsprogramm mehrere Aufträge parallel abzusetzen.

Dies macht bei *einem* PROFIdrive-Antrieb keinen Sinn, kann aber bei Kommunikation zu *mehreren* Antrieben parallel (oder ggf. bei anderen Geräten, die das unterstützen) sinnvoll sein.

Bei SIMOTION werden Ressourcen vorgehalten, um max. 8 Aufrufe von _write/_readRecord verwalten zu können. Die Unterscheidung erfolgt anhand der commandID von _write/_readRecord. Wird versucht, einen neunten Aufruf gleichzeitig abzusetzen, so wird dies mit Fehler von der Steuerung quittiert und unterbunden.

Der zeitliche Ablauf kann der Abbildung **8 Aufträge gleichzeitig verwalten** entnommen werden.

Es werden zunächst sieben Aufträge _writeRecord angestoßen, aber nicht beendet (keine weiteren Aufrufe von _writeRecord, um die Aufträge zum Abschluss zu bringen). Der achte Auftrag _writeRecord wird angestoßen und weiter bearbeitet bis zum Abschluss. Anschließend ist es möglich, einen neunten Aufruf abzusetzen (der hier aber wieder vom Anwenderprogramm nicht weiter bearbeitet wird). Die SIMOTION Systemfunktion _writeRecord quittiert dann den Versuch den zehnten Auftrag abzusetzen mit Fehler 16#80C3, da dies der neunte "offene" Auftrag wäre.

Hinweis

Die Obergrenze gilt pro SIMOTION Steuerung, nicht pro Bus-Segment an der Steuerung. Es ist also unerheblich, ob die adressierten Zielgeräte an einem PROFIBUS-Segment betrieben werden, oder auf mehrere verteilt sind.

Hinweis

Da die Systemfunktionen _write/_readRecord universell einsetzbar sind, wird auf Steuerungsseite *nicht* verriegelt, dass nur ein "Datensatz Schreiben/Lesen" pro PROFIdrive-Antrieb zu einer Zeit angestoßen werden kann.

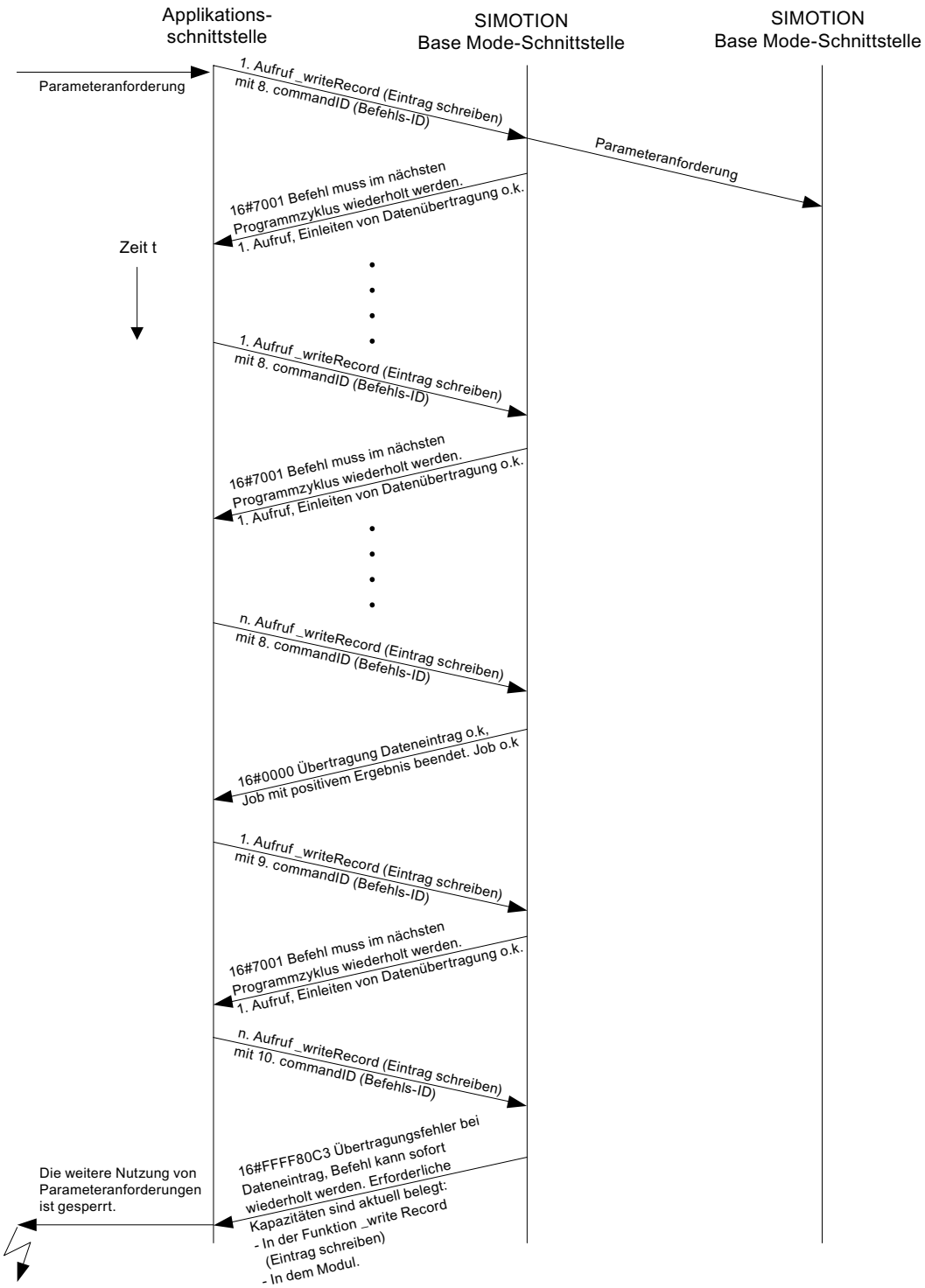


Bild 3-7 8 Aufträge gleichzeitig verwalten

Hinweis

Wenn Fehler 16#80C3 auftritt, müssen Sie die CPU auf STOP und dann wieder auf RUN setzen. Dadurch wird der Auftragspuffer gelöscht. Um den Fehler zu vermeiden, sollten Sie den Auftrag durch einen Abort-Befehl beenden, falls Sie den Auftrag nicht abschließen können.

3.6.9 Regeln für SIMOTION Befehle _writeDrive.../_readDrive...

3.6.9.1 Gültigkeitsbereich für die Regeln

Beschreibung

Die folgenden Beispiele werden am Beispiel der Systemfunktion _readDriveParameter dargestellt. Die Beschreibungen gelten aber analog auch für die anderen bereits erwähnten Systemfunktionen _writeDrive.../_readDrive... .

3.6.9.2 Regel 6 - Systemfunktion bei asynchroner Programmierung wiederholt aufrufen

Beschreibung

Der Anwender muss bei asynchroner Programmierung die Systemfunktion wiederholt aufrufen mit gleichen IDs, bis zur Beendigung der Funktion ("Langläufer").

In der folgenden Abbildung wird die fehlerfreie Verwendung der Systemfunktion _readDriveParameter dargestellt.

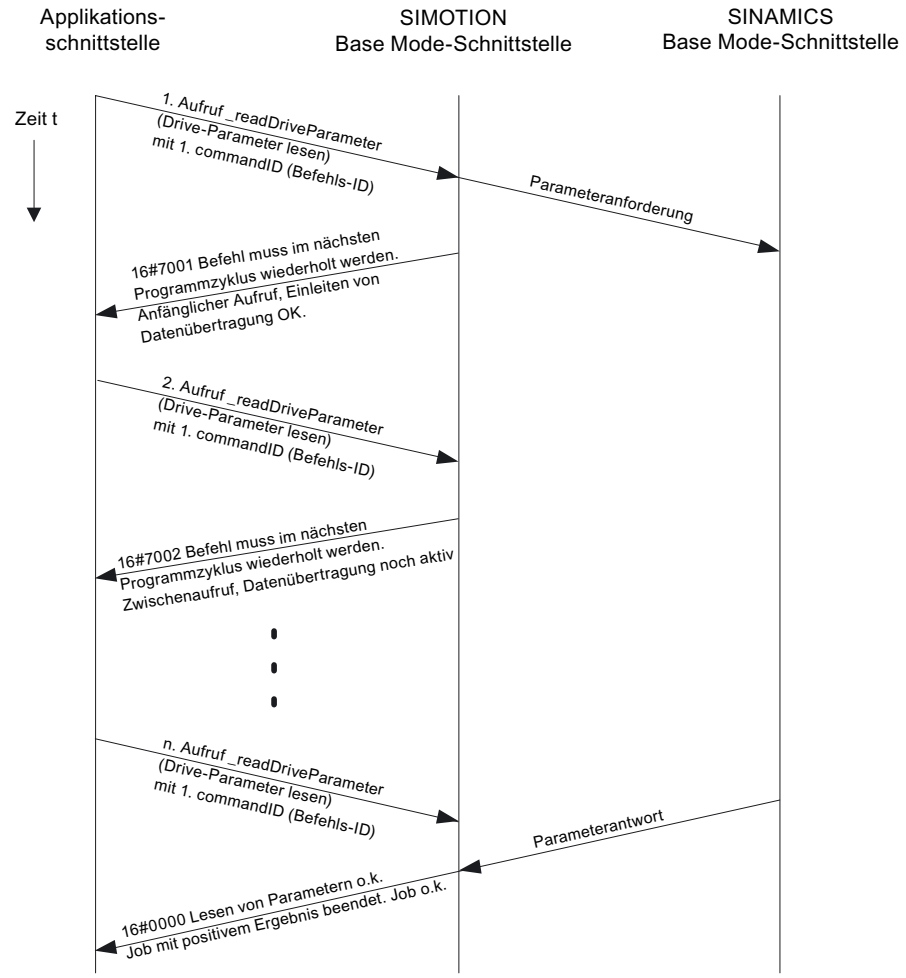


Bild 3-8 Fehlerfreier Ablauf mit den Systemfunktionen _writeDriveParameter und _readDriveParameter

3.6.9.3 Regel 7 -- Mehrere Aufrufe pro Zielgeräte gleichzeitig verriegeln

Beschreibung

In der PROFIdrive Norm ist festgelegt, dass PROFIdrive-Antriebe kein Pipelining durchführen und daher nur ein Auftrag zu einer Zeit bearbeitet wird. Dies ist daher auch für SINAMICS S120 im Inbetriebnahmehandbuch zu SINAMICS S120 dokumentiert.

Da die SIMOTION Systembefehle _write/_readDrive... für die häufig vorkommende Verwendung mit PROFIdrive-Antrieben erstellt wurden, wird dies bereits auf Steuerungsseite berücksichtigt.

Hinweis

Es ist dabei unerheblich, über welche Systemfunktionen die Übertragung in der Steuerung abgewickelt wird. Ein PROFIdrive-Antrieb kann nur einen Auftrag zu einer Zeit bearbeiten.

Folge für die Applikation auf der Steuerung:

Es muss verriegelt werden, dass die Applikation bzw. unterschiedliche Teile der Applikation gleichzeitig bzw. überlappend Aufträge an dasselbe PROFIdrive-Antriebsgerät senden.

Die Abbildung unten zeigt das Verhalten, wenn dies nicht berücksichtigt wird. Der Versuch, einen zweiten Auftrag (mit eigener commandID) an dasselbe Zielgerät abzusetzen, wird mit Fehler quittiert. Ein weiterer Auftrag an dasselbe Zielgerät kann erst dann wieder abgesetzt werden, wenn der erste Auftrag beendet bzw. abgebrochen wurde, siehe Abschnitt Freigabe der Verriegelung (Seite 61).

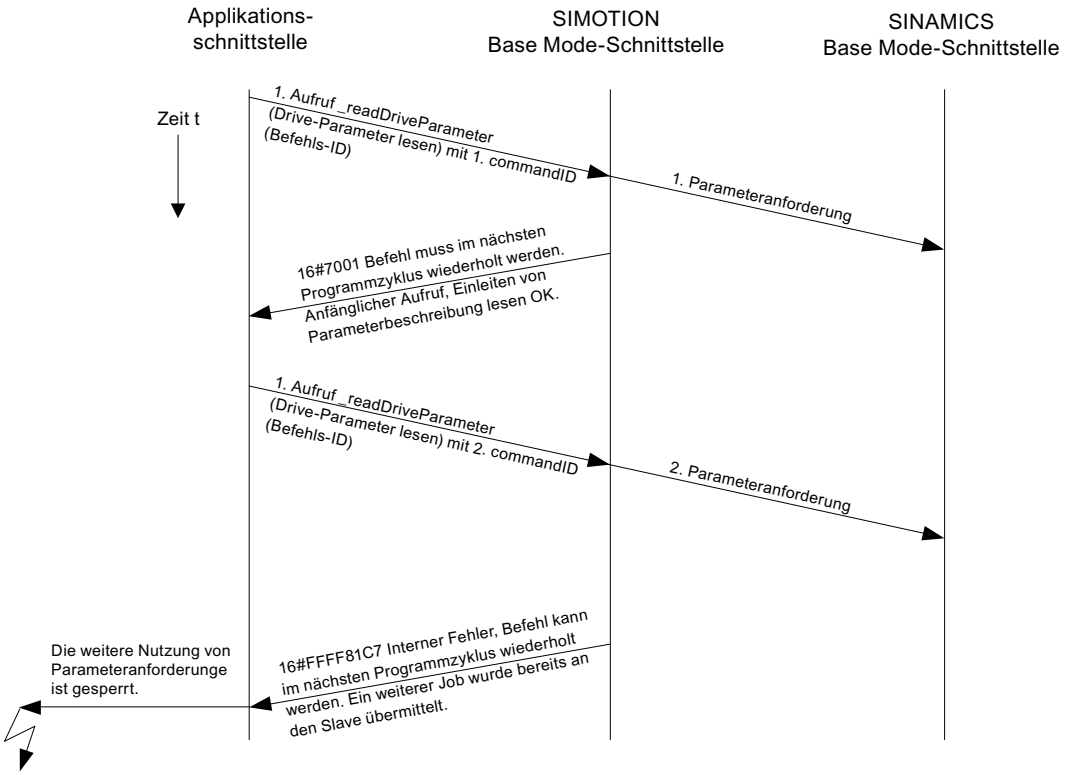


Bild 3-9 Verriegelung von mehreren Aufträgen _readDriveParameter auf ein Zielgerät

3.6.9.4 Regel 8 - Freigabe der Verriegelung nach vollständiger Abarbeitung eines Auftrags

Freigabe der Verriegelung erst bei vollständigen Abarbeitung eines Auftrags

Die folgende Abbildung zeigt, dass es nicht ausreicht, "etwas" zu warten, sondern dass man die Systemfunktionen _read/_writeDrive... wirklich so lange aufrufen muss, bis der Auftrag

vollständig abgearbeitet ist. Vorher wird die Verriegelung nicht gelöst und die internen Verwaltungsressourcen frei gegeben.

Die Anzahl der Aufrufe wurde so gewählt, dass die SIMOTION DP-V1 Schnittstelle jeden Folgeaufruf für den ersten Auftrag mit 16#7002 beantwortet und somit nicht vollständig bearbeitet wird. Dies kann je nach Belastung des Busses und des Antriebs auch recht häufig erforderlich sein (>25 mal). Eine Abschätzung dafür kann nicht angegeben werden.

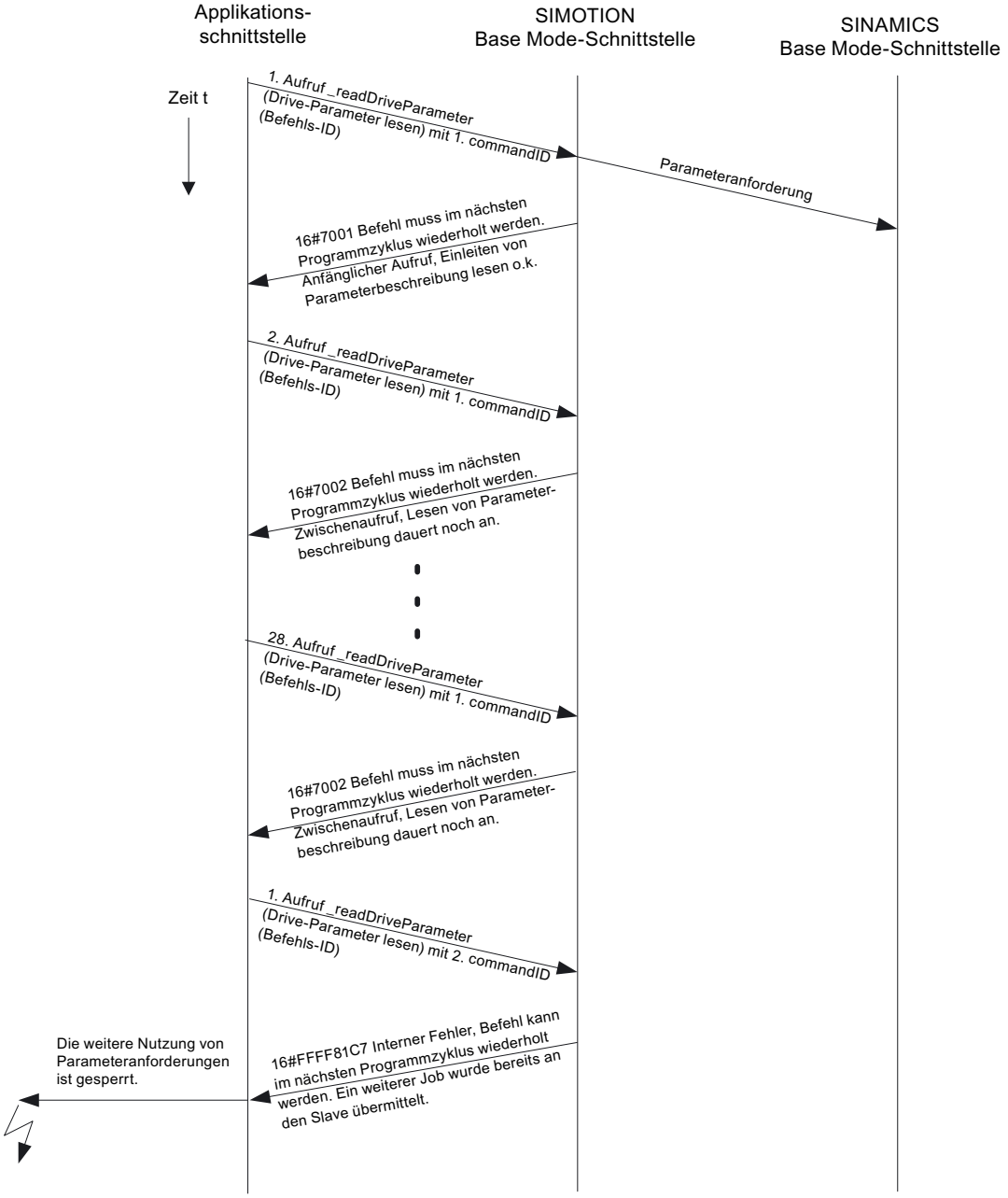


Bild 3-10 Vollständige Abarbeitung eines _readDriveParameters erforderlich zur Freigabe der Verriegelung

3.6.9.5 Regel 9 - Abbrechen von Aufträgen bei asynchronem Aufruf

CommandID wird benötigt zum Abbrechen von Aufträgen bei asynchronem Aufruf

Um den DP-V1 Dienst für das Zielgerät wieder frei zugeben, muss

- entweder der erste Auftrag beendet werden (wiederholte Aufrufe mit der commandID des ersten Auftrags)
- oder abgebrochen werden (nochmals ein Aufruf der Funktion _readDriveParameter mit der gleichen commandID wie beim ersten Anstoßen des Auftrags auch. Zusätzlich wird dazu der Eingabeparameter nextCommand mit dem Wert ABORT_CURRENT_COMMAND versehen).

Hinweis

Ab V4.1 ist ein Abbrechen ohne Kenntnis der commandID möglich, siehe Löschen von _readDrive- und _writeDrive-Aufträgen (Seite 52) .

Ein exemplarischer Aufruf der Funktion _readDriveParameter mit der ersten commandID (id1) und ABORTED_CURRENT_COMMAND sieht wie folgt aus:

```
Return_Par_read_delete :=
    readDriveParameter(
    ioId:=INPUT,
    logAddress := 256,
    parameterNumber := number,
    numberOfElements := 0,
    subIndex:= 0,
    nextCommand :=
    ABORT_CURRENT_COMMAND,
    commandId := id1);
```

Die zeitliche Abfolge kann der folgenden Abbildung entnommen werden.

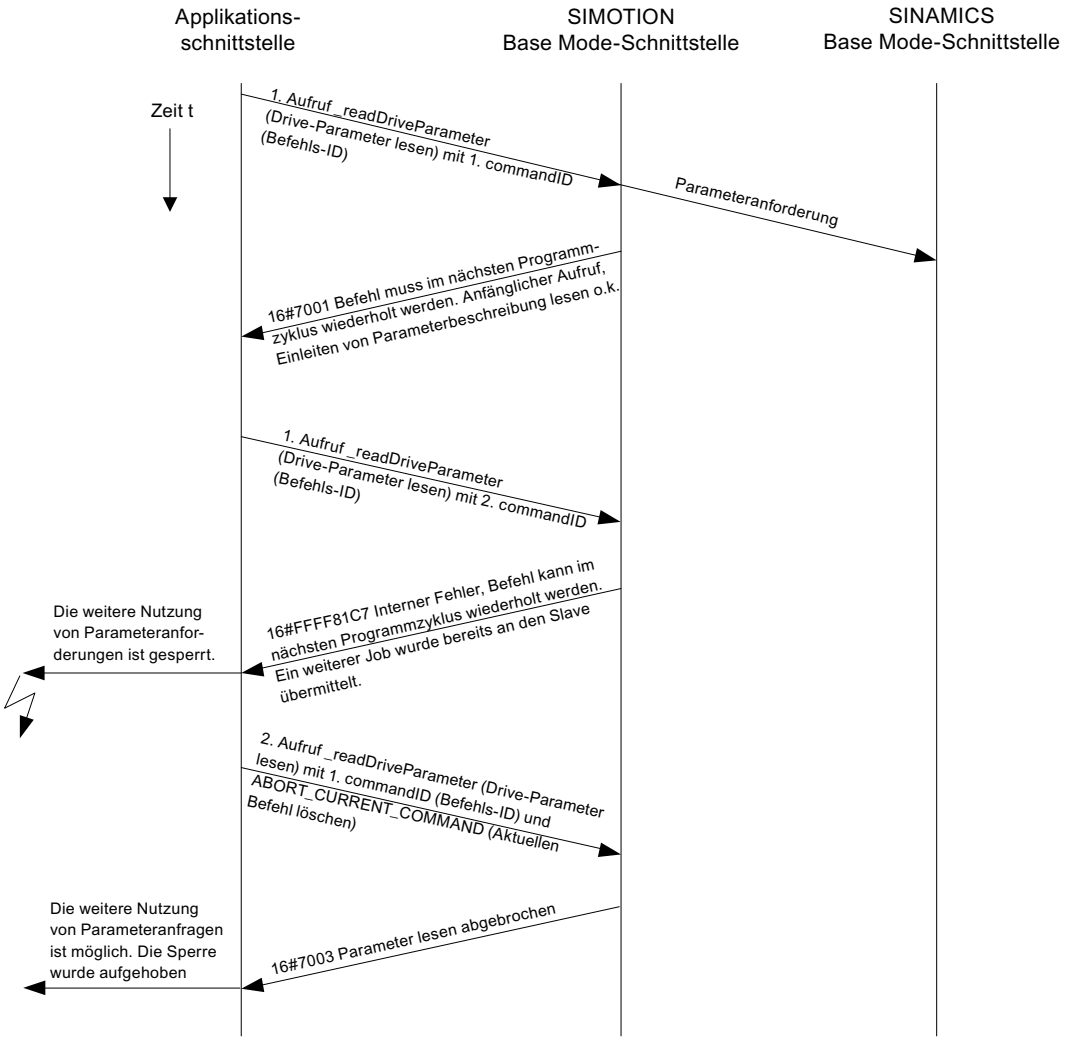


Bild 3-11 Abbrechen eines Auftrags _readDriveParameter mit bekannter commandID

Der in der folgenden Abbildung dargestellte Ablauf zeigt, dass es nicht gelingt, einen Auftrag abzubrechen ohne Kenntnis der ursprünglichen commandID. Es wird der Abbruchversuch abgebrochen, nicht der erste Auftrag. Der Grund dafür ist, dass die commandID zur Verwaltung der unterschiedlichen Aufträge im System herangezogen wird.

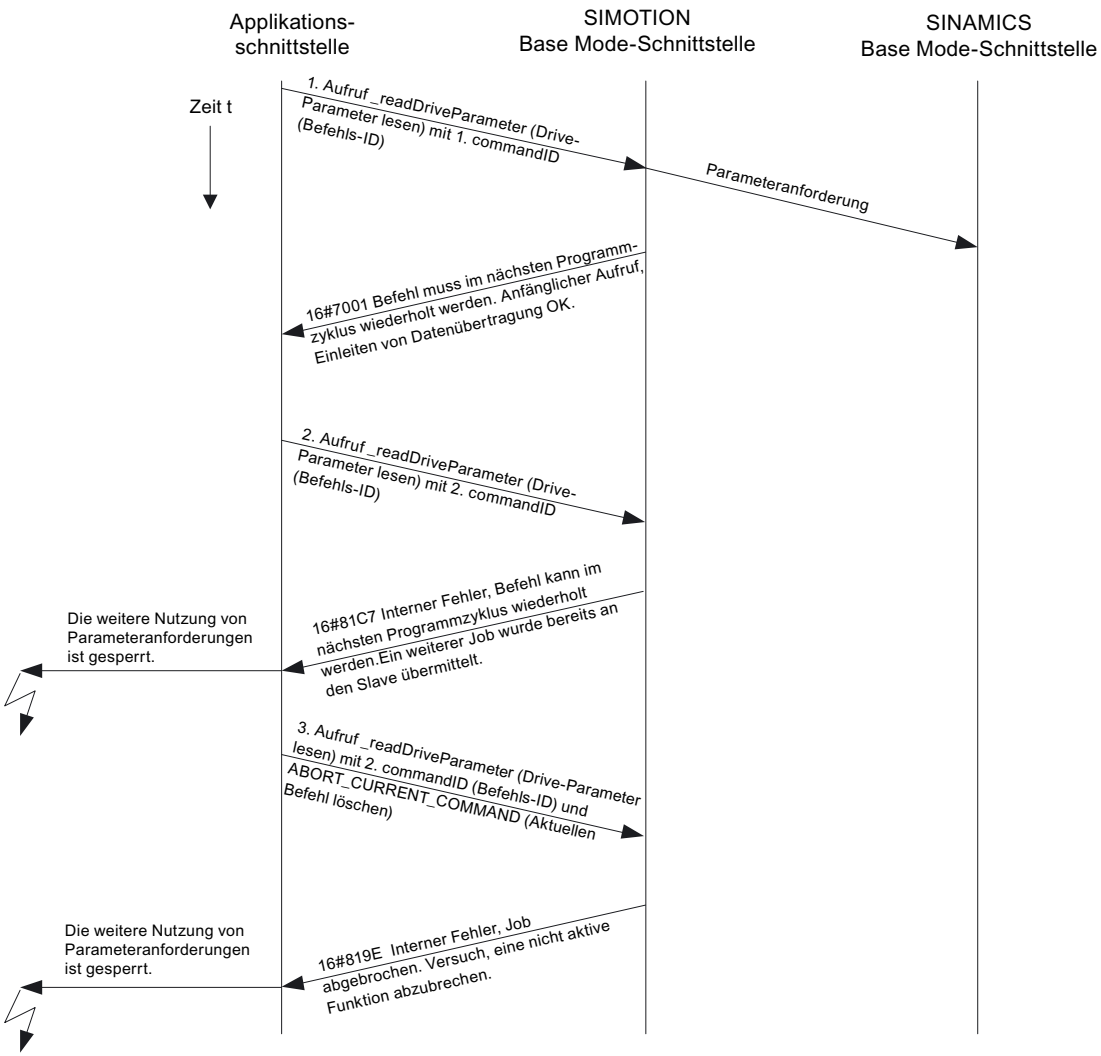


Bild 3-12 Kein Abbrechen eines Auftrags _readDriveParameter mit neuer CommandID

Hinweis

Wichtig ist also, dass das Anwenderprogramm die commandID der Aufträge so lange behält, bis der Auftrag beendet oder abgebrochen ist.

Hinweis

Besonders ist darauf zu achten, dass – z.B. durch sonstige Bedingungen gesteuert – im Anwenderprogramm nicht die Bearbeitung der _write/_readDrive... Funktionen übersprungen wird, solange diese noch nicht beendet sind.

3.6.9.6 Regel 10 - Verwaltung von 16 Aufträgen

SIMOTION verwaltet max. 16 Aufrufe für verschiedene Geräte parallel

In der Steuerung stehen begrenzte Ressourcen (Speicherplatz) zur Speicherung von Verwaltungsdaten für _write/_readDrive... Systemfunktionsaufrufe zur Verfügung. Werden zu viele Aufrufe parallel abgesetzt, so erfolgt eine Fehlermeldung, analog zur Grenze bei _read/_writeRecord in Abschnitt Maximale Anzahl von Aufrufen (Seite 57).

Bei SIMOTION werden Ressourcen vorgehalten, um max. 16 Aufrufe von _writeDrive.../_readDrive....-Systembefehlen verwalten zu können. Die Unterscheidung erfolgt anhand der commandID. Wird versucht, einen siebzehnten Aufruf gleichzeitig abzusetzen, so wird dies mit Fehler von der Steuerung quittiert und unterbunden.

3.6.9.7 Regel 11 - Parallele Aufträge unterschiedlicher Antriebsgeräte

Parallele Aufträge an unterschiedlicher Antriebsgeräte sind möglich

Die **Parallele Bearbeitung von _readDriveParameter-Aufträgen** Abbildung zeigt, dass parallel Aufträge mit verschiedenen Antriebsgeräten abgewickelt werden können. Der SINAMICS Integrated einer D445 (z.B.) wird als erstes PROFIdrive-Antriebsgerät und die Erweiterungsbaugruppe CX32 als zweites PROFIdrive Antriebsgerät von einer Steuerung SIMOTION D445 genutzt.

Insgesamt werden in dem Beispiel drei Leseaufträge (zwei Aufträge auf das erste Antriebsgerät (SINAMICS Integrated) und ein Auftrag an das zweite Antriebsgerät (CX32) mittels der Systemfunktion _readDriveParameter abgesetzt.

- Der erste Leseauftrag für die SINAMICS Integrated wird absichtlich nur einmal aufgerufen, so dass die Verriegelung anspricht.
- Anschließend wird der zweite Leseauftrag auf das zweite PROFIdrive-Antriebsgerät (CX32) abgesetzt. Dieser Auftrag wird erfolgreich abgearbeitet.
- Der dritte Leseauftrag ist wieder an das erste Antriebsgerät (SINAMICS Integrated) adressiert und kann wegen des noch laufenden ersten Auftrags nicht mehr erfolgreich ausgeführt werden.

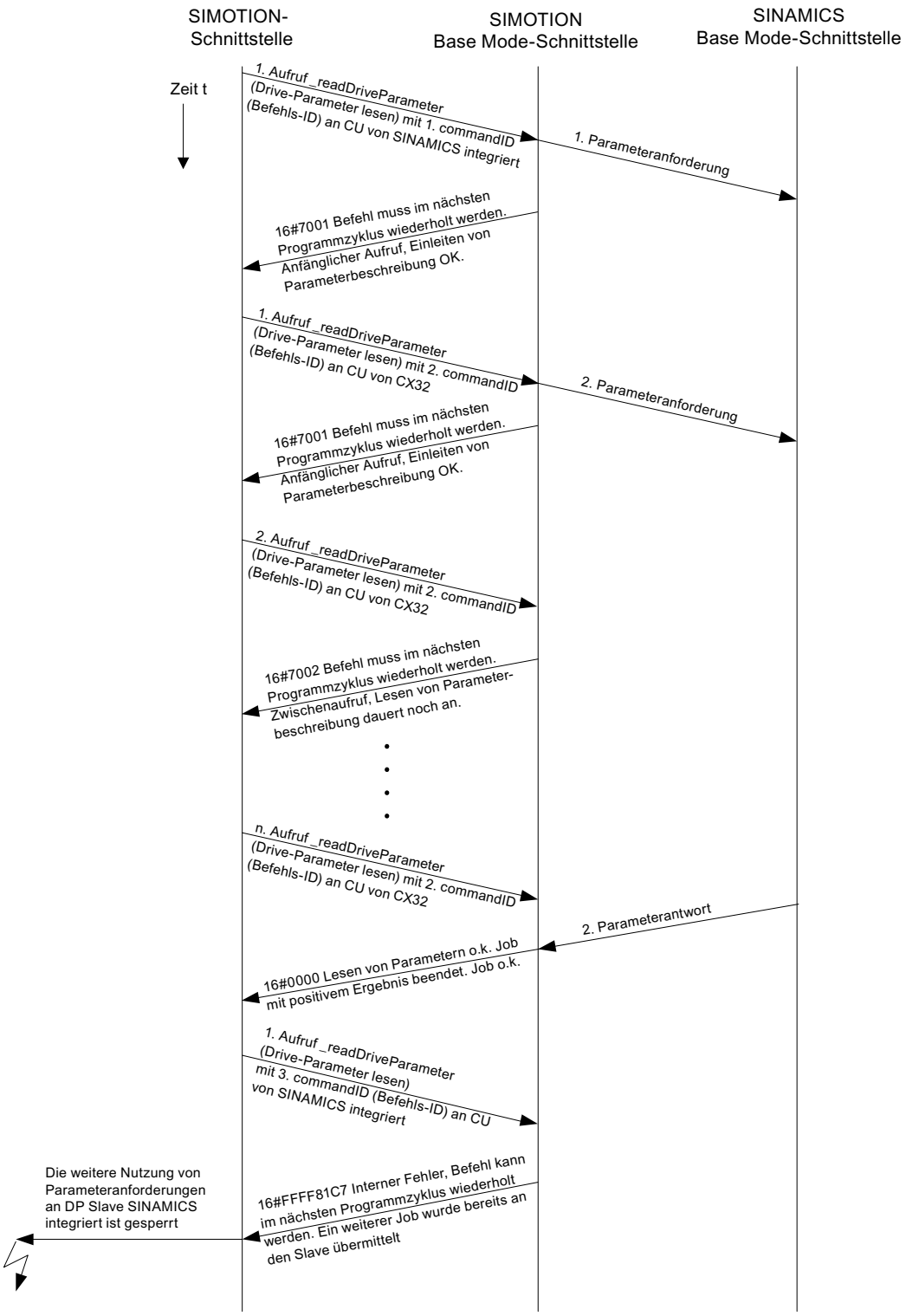


Bild 3-13 Parallele Bearbeitung von _readDriveParameter-Aufträgen an unterschiedlichen Antriebsgeräten einer Steuerung

3.6.10 Besonderheiten

3.6.10.1 Regel 12 - Datenpufferung von maximal 64 Antriebsobjekten

SIMOTION puffert die Daten von max. 64 Antriebsobjekten´

Der erste Aufruf der Funktionen _write/_readDrive... nach Systemhochlauf läuft deutlich länger als folgende Aufrufe zum gleichen Antriebsobjekt.

- Das System muss erstmalig interne Verwaltungsinformationen aufbauen, die für folgende Aufrufe zum selben Antriebsobjekt schneller zugegriffen werden können.

Es können in SIMOTION die Daten für bis zu 64 Antriebsobjekte gespeichert werden für die Verwendung mit _write/_readDrive... Dabei wird die Unterscheidung anhand der I/O-Adresse vorgenommen.

3.6.10.2 Regel 13 - Systemfunktionen können gemischt verwendet werden

Systemfunktionen _writeRecord/_readRecord und _writeDrive.../_readDrive... können gemischt verwendet werden

Eine gemischte Verwendung folgender Systembefehle ist grundsätzlich möglich:

- SIMOTION Systembefehle _writeRecord/_readRecord
- SIMOTION Systembefehle _writeDrive.../_readDrive...

Hinweis

Wichtig ist aber, dass man sich klarmacht, dass damit bei fehlender Verriegelung der Systembefehle aus beiden Befehlsgruppen mehrere Aufträge an einen PROFIdrive-Antrieb abgesetzt werden könnten (siehe folgender Abschnitt), was ein PROFIdrive-Antrieb nicht verarbeiten kann. Die systeminterne Verriegelung von _write/_readDrive... wird so umgangen.

In der Abbildung **Gemischte Verwendung von _readDrive... und _read/_writeRecord** ist dargestellt, dass insbesondere die Funktionen _write/_readRecord auch dann noch zum selben Zielgerät verwendet werden können, wenn wegen eines noch laufenden Auftrags _readDriveParameter weitere Aufträge mit demselben Befehl vom System unterbunden werden – dies ist vom Anwender zu verriegeln, da ein PROFIdrive-Antrieb dies nicht verarbeiten kann.

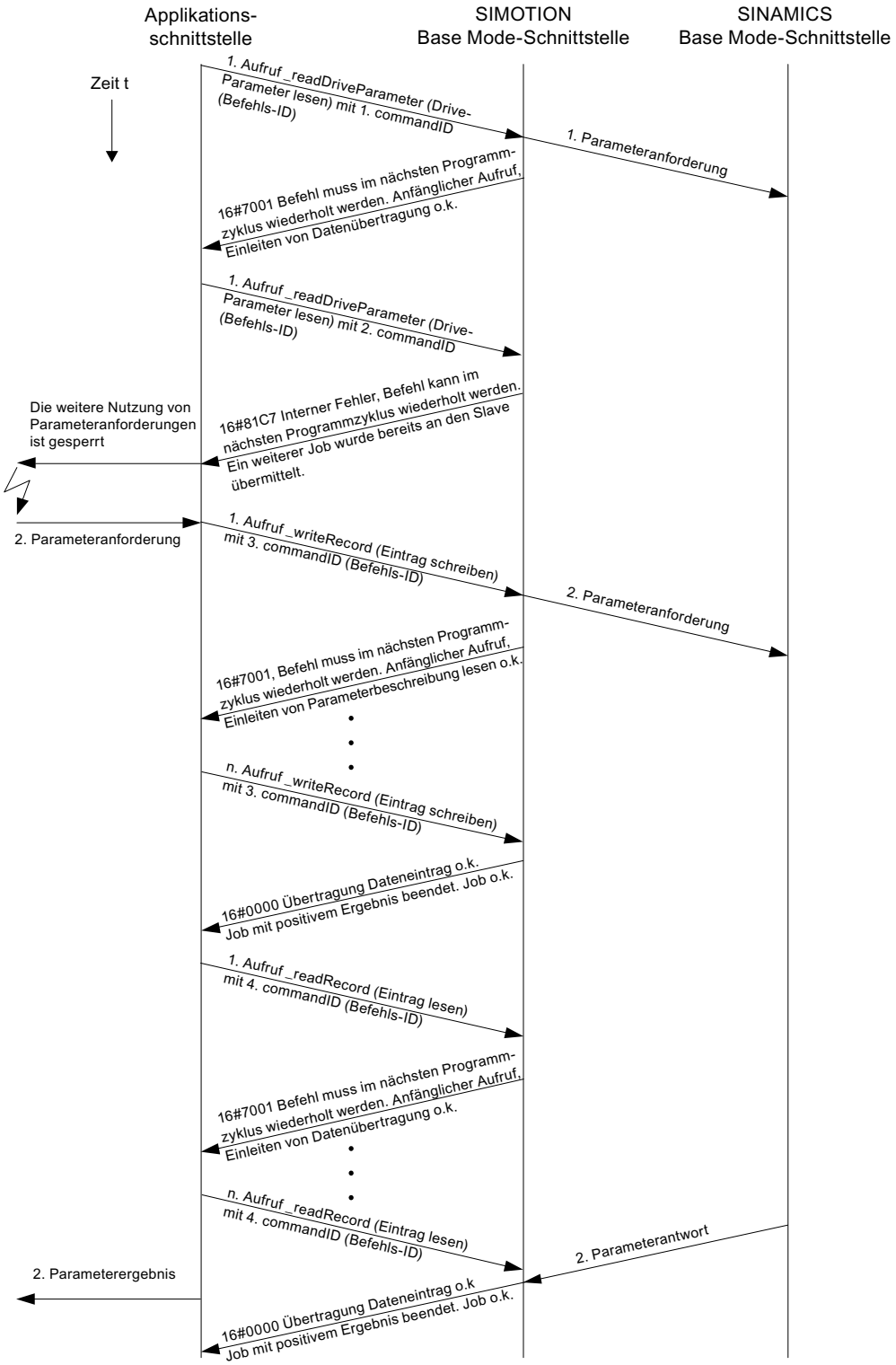


Bild 3-14 Gemischte Verwendung von _readDrive... und _read/_writeRecord

3.6.10.3 Regel 14 - Verriegelung bei gemischter Verwendung der Befehle

Anwender muss verriegeln bei gemischter Verwendung der Befehle aus den beiden Befehlsgruppen

Bei gemischter Verwendung der folgenden Systembefehle kann es dazu kommen, dass zu einem Gerät mehr als ein "Datensatz Lesen/Schreiben" gleichzeitig abgesetzt wird, da bei SIMOTION nur innerhalb der Befehlsgruppen verriegelt bzw. gepuffert wird, aber nicht untereinander.

- SIMOTION Systembefehle _writeRecord/_readRecord

und

- SIMOTION Systembefehle _writeDrive.../_readDrive...

Dies ist vom Anwender ggf. zu verriegeln, um Datenverlust/Überschneidungen zu vermeiden, da ein PROFIdrive Antrieb laut PROFIdrive Profil kein Pipelining durchführt und daher nur einen Auftrag pro Zeit bearbeiten kann.

3.6.11 Programm-Beispiele

3.6.11.1 Programmbeispiel

Beschreibung

Das nachfolgende Beispiel zeigt, wie die Systembefehle _writeRecord bzw. _readRecord verwendet werden können, um den Störcode aus Parameter p0945 eines SINAMICS-Antriebs (Antriebsobjekt DO3, I/O-Adresse 256) auszulesen.

Beispiel

Das Beispiel-Programm kann z.B. in der BackgroundTask aufgerufen werden, da die sog. "asynchrone Programmierung" verwendet wird.

```
//=====================================================================
// demonstrate reading parameter 945 (fault code) via data set 47
// using SIMOTION system functions _write/_readRecord (asynchronous call)
// INPUT address 256 is assumed to address the SINAMICS
// drive is DO3 in SINAMICS S120
//=====================================================================
INTERFACE
 PROGRAM record;
 // declare request type
 TYPE
 // declare struct of header request
 Header_Type_Request : STRUCT
    Request_Reference : USINT;
    Request_Id : USINT;
```

```
    Axis : USINT;
    Number_Of_Parameter : USINT;
 END_STRUCT;

 // declare struct of parameter address request
 Parameter_Address_Request : STRUCT
    Attribute : USINT;
    Number_Of_Elements : USINT;
    Parameter_Number : UINT;
    SubIndex : UINT;
 END_STRUCT;

 // declare struct of request
 Request : STRUCT
    Header : Header_Type_Request;
    ParameterAddress : Parameter_Address_Request;
 END_STRUCT;
// declare struct of header response
 Header_Type_Response : STRUCT
 Response_Reference : USINT;
 Response_Id : USINT;
 Axis : USINT;
 Number_Of_Parameter : USINT;
 END_STRUCT;

 // declare struct of parameter address response
 Parameter_Address_Response : STRUCT
 Format : USINT;
 Number_Of_Elements : USINT;
 Value_Or_Error_Value : DWORD; // dependent on format
 END_STRUCT

 // declare struct of response
 Response : STRUCT
 Header : Header_Type_Response;
 ParameterAdress : Parameter_Address_Response;
 END_STRUCT;
 END_TYPE
 // declare global variables
 VAR_GLOBAL
 // declare variable, that represents the dataset 47 request
 myRequest : Request;
 // declare variable, that represents the dataset 47 response
 myResponse : Response;
 // declare variable, that returns a value after calling _writeRecord
 myRetDINT : DINT;
 // declare variable, that returns a struct after calling _readRecord
 myRetstructretreadrecord : StructRetReadRecord;
 // declare array of byte,
 // which helps to create the request/response
 // with marshalling function
```

```
 bytearray : ARRAY[0..239] OF BYTE;
 // declare array of USINT,
 // because the systemfunctions _writeRecord and _readRecord
 // use this array
 usintarray : ARRAY[0..239] OF USINT;
 // declare command ids
 id_write, id_read : commandidtype;
 // declare the variable, to control step by step execution
 // start cycle with setting to 0 by user
 program_step : USINT := 3; // initially idle;
 END_VAR
END_INTERFACE
```

Implentation

```
// ===============================================================
IMPLEMENTATION
PROGRAM record
CASE program_step OF
// initialize -----------------------------------------------------
0:
 // get command ids for calling system functions
 id_write := _getcommandid();
 id_read := _getcommandid();
 // header from the request
 // here: Axis-No / DO-ID is 3
 // read Parameter 945 (drive fault code)
 myRequest.Header.Request_Reference := 16#10; // arbitrary no.
 myRequest.Header.Request_Id := 16#1; // read request
 myRequest.Header.Axis := 16#3; // axis no 3
 myRequest.Header.Number_Of_Parameter := 16#1; // one parameter

 // parameter address from the request
 myRequest.ParameterAddress.Attribute := 16#10; // read value
 myRequest.ParameterAddress.Number_Of_Elements := 16#1; // one index
 myRequest.ParameterAddress.Parameter_Number := 945; // parameter no.
 myRequest.ParameterAddress.SubIndex := 0;

 // convert myRequest to a BIBBYTEARRAY to use the marshalling functions
 // two step conversion from user defined data type
 // to usintarray type required by system functions
 bytearray := ANYTYPE_TO_BIGBYTEARRAY(myRequest,0);
 usintarray := BIGBYTEARRAY_TO_ANYTYPE(bytearray,0);

 // next step
 program_step := 1;

// execute _writeRecord -------------------------------------------
1:
```

```
 // the systemfunctions _writeRecord and _readRecord
 // have to be called in sequence.
 // the functions occur always as pair.
 // call systemfunction _writeRecord to send the request
 myRetDINT := _writerecord(
    ioid := INPUT,
    logaddress := 256, // io address
    recordnumber := 47, // data set 47 for DPV1
    offset := 0,
    datalength := 240,
    data := usintarray, //
    nextcommand := IMMEDIATELY, // use asynchronous
    commandid := id_write // use known commandID
 );
 // check the return value
 // keep calling until _writeRecord ready
 IF(myRetDINT = 0)THEN
    // next step
    program_step := 2;
 END_IF;
// wait for requested data ------------------------------------------------
// execute _readRecord
2:
 // call systemfunction _readRecord to receive the data
 myRetstructretreadrecord := _readrecord(
    ioid := INPUT,
    logaddress := 256, // io address
    recordnumber := 47, // data set 47 for DPV1
    offset := 0,
    datalength := 240,
    nextcommand := IMMEDIATELY, // use asynchronous
    commandid := id_read // use known commandID
 );
 // check the return value
 // keep calling until _readRecord ready
    IF(myRetstructretreadrecord.functionresult = 0)THEN
        // next step
        program_step := 3; // --> done
        // get data
        // two step conversion into user defined data type
        // from usintarray type given by system functions
        bytearray :=ANYTYPE_TO_BIGBYTEARRAY(
        myRetstructretreadrecord.data,0);
        myResponse := BIGBYTEARRAY_TO_ANYTYPE(bytearray,0);
        // received data can now be read from myResponse...
    END_IF;
END_CASE;
END_PROGRAM
END_IMPLEMENTATION
```

PROFIsafe

4

4.1 Kommunikationsbeziehungen bei Drive Based Safety

Beschreibung

Die Drive Based Safety Funktionen im Antrieb können entweder über sichere Klemmen direkt am Antrieb oder über Profibus/Profinet von einer fehlersicheren Steuerung (F-Steuerung) aus gesteuert werden.

Die Ansteuersignale zu den Drive Based Safety Funktionen sowie die Rückmeldesignale zum Safety Funktionsstatus sind sicherheitsrelevant und müssen deshalb über einen, mit dem PROFIsafe Protokoll gesicherten, Kommunikationskanal übertragen werden. Das allgemeine Scenario der Wechselwirkung zwischen den verschiedenen Steuerungs- und Antriebsprozessen sowie die hierfür notwendigen Kommunikationsbeziehungen zwischen diesen zeigt schematisch das folgende Bild.

Signalfluss für die An- und Abwahl der Drive Basedd Safety Funktionen
und deren Signalisierung an den Antriebsteuerungsprozess

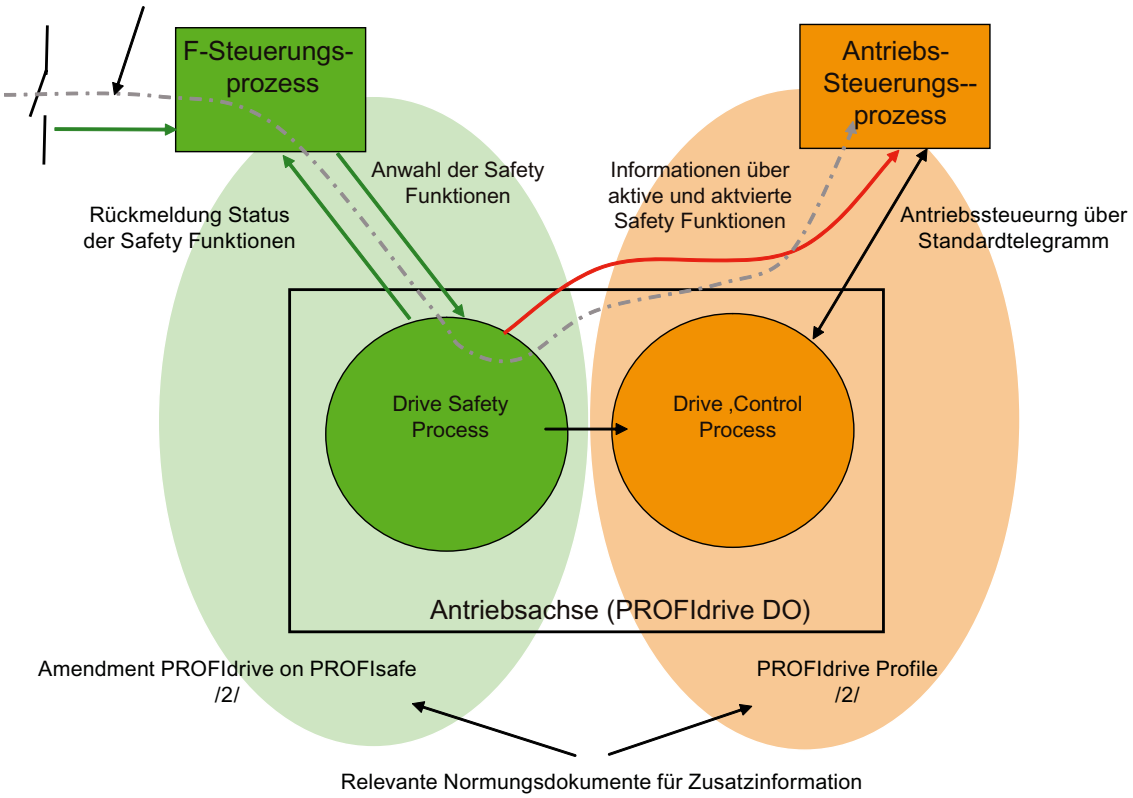


Bild 4-1 Kommunikationsbeziehungen bei Drive Based Safety

Die Schnittstellen des "Drive Safety Prozess" zur F-Steuerung und zur Antriebs-Steuerung ist eine PROFIdrive Schnittstelle und deren Funktionalität in /1/ und /2/ definiert.

Die Einleitung und Überwachung einer Drive Based Safety Funktion erfolgt durch die F-Steuerung über den mit PROFIsafe gesicherten Übertragungskanal von der F-Steuerung zum Antrieb (Antriebsachse). Die jeweiligen Zustände der Drive Based Safety Funktionen im Antrieb haben auch Rückwirkungen auf die Antriebsschnittstelle zur Antriebssteuerung, da bei einigen Drive Based Safety Funktionen die Antriebshoheit zwischen Antriebssteuerung und Drive Safety Process wechselt.

Zur effektiven Koordination von F-Steuerung und Antriebssteuerung ist deshalb zusätzlich ein Informationskanal vom "Drive Safety Process" zur Antriebssteuerung notwendig, damit die Antriebssteuerung auf die gewünschten bzw. aktivierten Drive Based Safety Funktionen entsprechend reagieren kann.

4.2 Telegramme und Signale bei Drive Based Safety

Beschreibung

Da Simotion über keine sichere Logikfunktion verfügt, kann der F-Steuerungsprozess nicht von Simotion wahrgenommen werden, sondern wird mit Hilfe eines zweiten Controllers mit F-Funktionalität, typischerweise eine SIMATIC F-CPU, realisiert. Folgendes Bild zeigt diese Konstellation am Beispiel einer Sinamics Achse. Vom Antriebs DO geht ein mit PROFIsafe gesicherter Kommunikationskanal zur SIMATIC F-CPU. Für diesen Kommunikationskanal steht in der Norm das Standardtelegramm 30 zur Verfügung, welches aus dem Safety Steuer- und Statuswort besteht. Durch das Anwenderprogramm auf der F-CPU werden über das Safety-Steuerwort die projektierten Drive Based Safety Funktionen im Antrieb (Drive safety process) an- oder abgewählt. Die Rückmeldung der aktiven Safety Funktionen erfolgt in den Input-Daten über das Safety-Zustandswort an die F-CPU. Die Rückmeldung der aktiven Safety Funktionen erfolgt von einem sicheren Prozess im Antrieb über einen gesicherten Kommunikationskanal und kann damit zur Freischaltung von Schutzräumen und Türen durch die F-CPU verwendet werden.

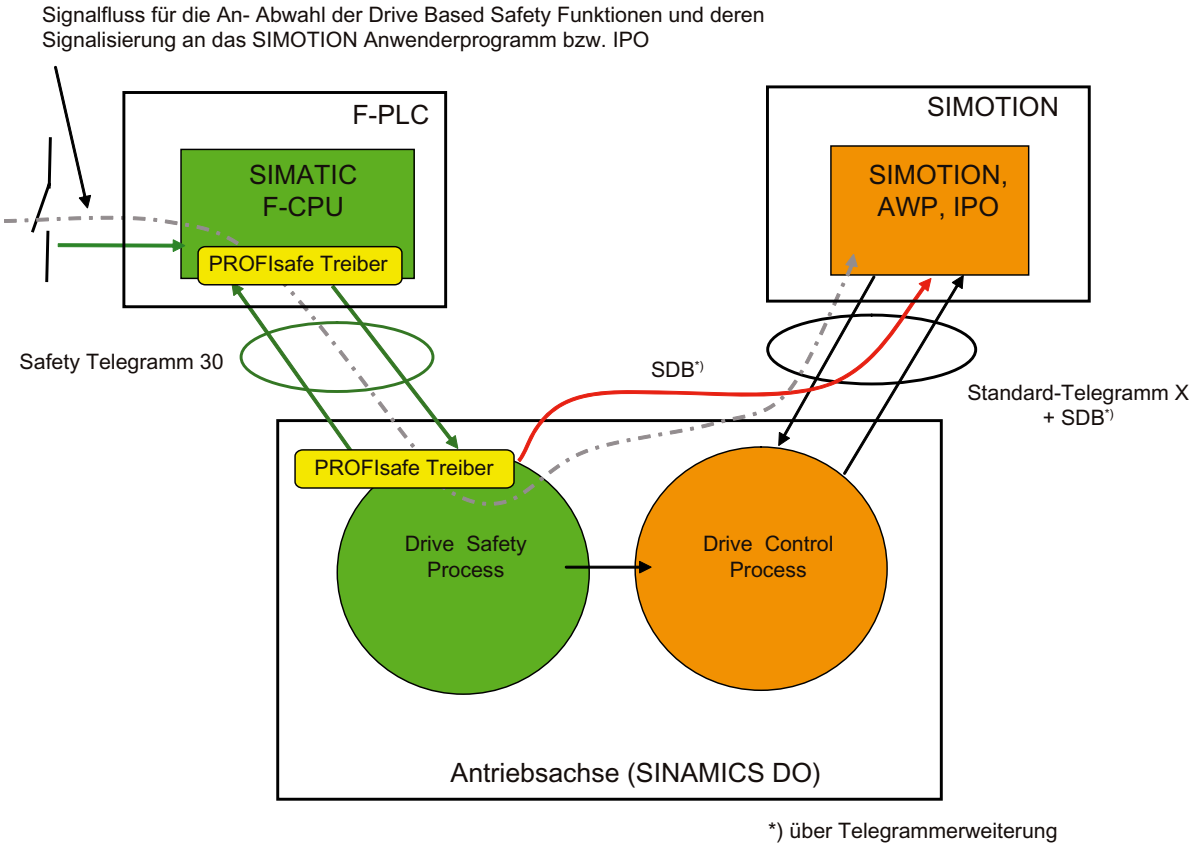


Bild 4-2 Telegramme und Signale bei Drive Based Safety

Parallel zur Steuerung der Safety Funktionen im Antrieb über den PROFIsafe Kanal, gibt es noch einen optionalen Safety Informationskanal (SDB), über den Aktivierungs- und Statuszustände der Drive Based Safety Funktionen vom Antrieb an SIMOTION übertragen werden. Dieser Informationskanal ist ungesichert und wird in der Praxis durch eine Telegrammverlängerung des Standardtelegramms um den SDB Datenblock realisiert. Zweck des Safety Informationskanals ist die optionale Einbindung der Antriebsbewegungssteuerung (IPO, SERO) sowie des gesamten Anwenderprogramms (AWP) in den höherprioren Ablauf der Drive Based Safety Funktionen sowie des Anwenderprogramms der F-Steuerung. Typische Simotion Reaktionen sind z.B.:

- Erkennen einer safetybedingten autonomen Handlung des Antriebes (z.B. Bremsrampe bei SS1 und SS2 und wechseln in den Nachführbetrieb.
- Erkennen einer Safety Funktionsanwahl und daraus abgeleiteter Simotion Reaktion zur Einleitung der Safety Funktion (z.B. steuerungsbasierte Bremsrampe bzw. Geschwindigkeitsabsenkung bei SOS und SLS).

4.3 F-Proxy Funktionen von SIMOTION

Beschreibung

Zur PROFIsafe Anbindung der integrierten Antriebe von Simotion D, sowie von SINAMICS Antrieben, die von Simotion gesteuert werden aber in einer anderen Kommunikationsdomain als die F-CPU liegen, verfügt Simotion über eine integrierte F-Proxy Funktionalität. Der F-Proxy ermöglicht ein transparentes Durchleiten von IO-Daten vom Simotion I-Slave bzw. I-Device Interface zum jeweiligen Simotion Master- bzw. Controller Interface an dem der Antrieb projektiert ist. Wie folgendes Bild zeigt, wird die Kommunikationsstrecke durch die PROFIsafe Treiber in der F-CPU und dem Antrieb als Ganzes gesichert.

Zur Nutzung der F-Proxy Funktionalität sind die beiden Kommunikationsstrecken, von der F-CPU zu Simotion und von Simotion zum Antrieb getrennt zu projektieren.

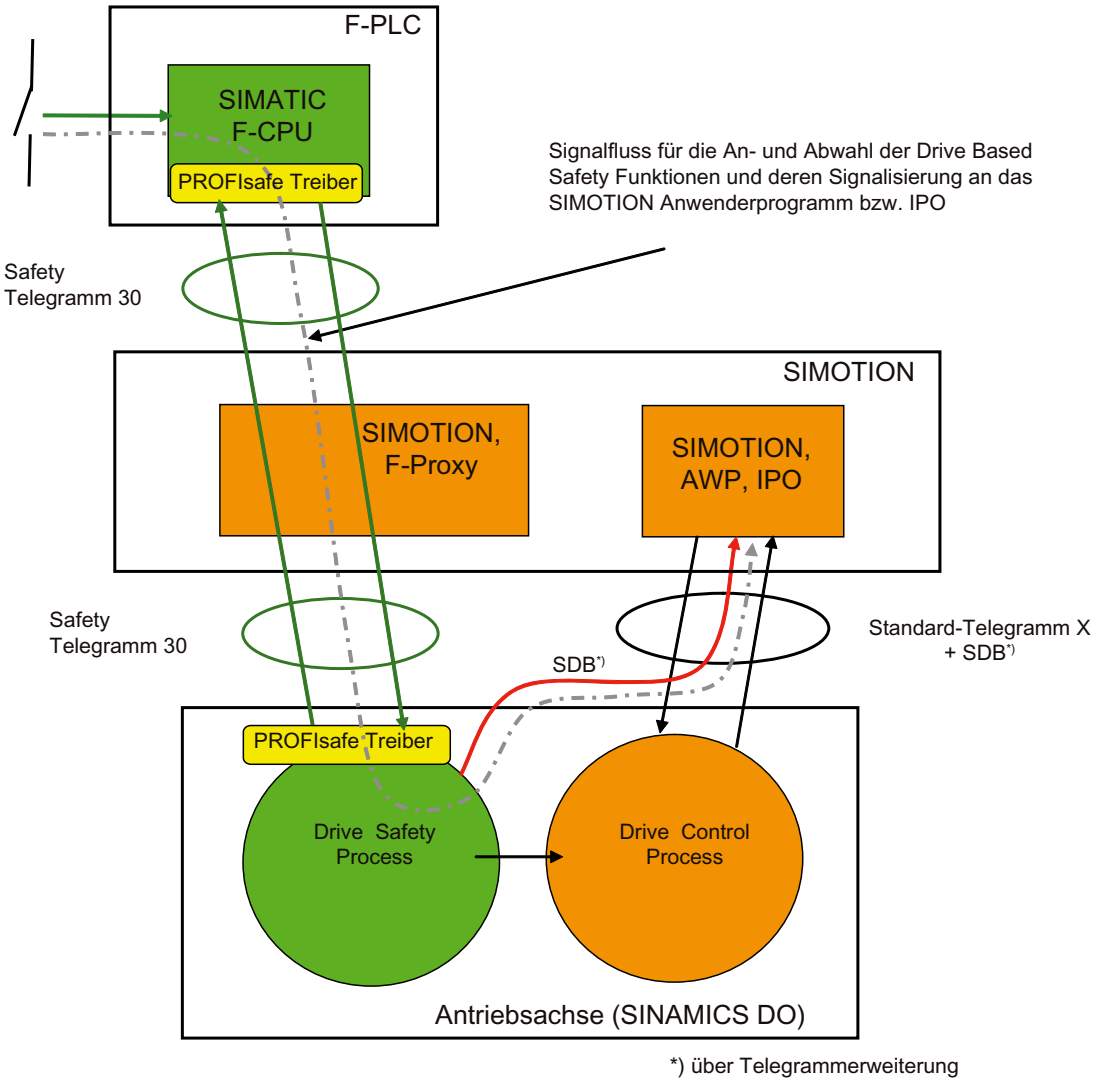


Bild 4-3 Durchleitung des PROFIsafe Kanals mit dem F-Proxy

4.4 Weitere Informationen zu SIMOTION und PROFIsafe

Beschreibung

Weitere Informationen zum Thema PROFIsafe erhalten Sie in folgenden Dokumentationen:

- Wie Sie eine Achse mit einem SINAMICS-Antrieb mit Safety Integrated verschalten, finden Sie beschrieben im Funktionshandbuch TO Achse / Externer Geber .
- Wie Sie einen SINAMICS S120 Antrieb bzw. einen SINAMICS S110 Antrieb mit Safety Integrated projektieren, finden Sie beschrieben in den Handbüchern
  - *Funktionshandbuch SINAMICS S120*
  - *Funktionshandbuch SINAMICS S120 Safety Integrated*
  - *Funktionshandbuch SINAMICS S110*.

PROFIBUS

5

5.1 PROFIBUS Kommunikation

5.1.1 PROFIBUS Kommunikation (Übersicht)

Beschreibung

PROFIBUS DP (Decentralized Peripherals) ist für den schnellen Datenaustausch in der Feldebene konzipiert. Die Kommunikation erfolgt zwischen einem PROFIBUS Master Klasse 1 (z.B. einem SIMOTION Controller) und PROFIBUS Slaves (z.B. einem SINAMICS S120 Antrieb). Der Datenaustausch mit den dezentralen Geräten erfolgt vorwiegend zyklisch (DP-V0-Kommunikation). Hierbei liest die zentrale Steuerung (SIMOTION Controller) die Eingangsinformationen zyklisch von den Slaves und schreibt die Ausgangsinformationen zyklisch an die Slaves. Weiter werden über die zyklischen Dienste Diagnosefunktionen zur Verfügung gestellt. Die folgende Grafik zeigt das Datenprotokoll am PROFIBUS DP.

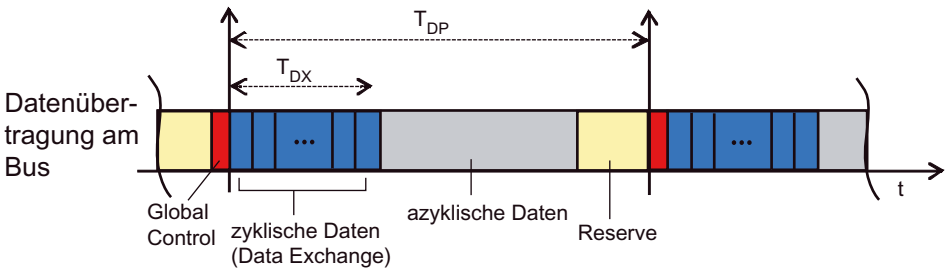


Bild 5-1 Datenprotokoll am Profibus

5.2 Kommunikation mit SIMATIC S7

5.2.1 Mögliche Kommunikationsverbindungen zwischen SIMOTION und SIMATIC

Im Folgenden werden die Möglichkeiten beschrieben, wie ein SIMOTION und ein SIMATIC S7 Gerät über PROFIBUS miteinander kommunizieren können.

Dazu gibt es verschiedene Möglichkeiten:

- Ein SIMOTION Gerät wird als DP-Slave an ein DP-Mastersystem einer SIMATIC S7 angebunden.
- Ein SIMATIC S7 Gerät wird als DP-Slave an ein DP-Mastersystem einer SIMOTION angebunden.
- Zwischen SIMOTION und SIMATIC S7 wird eine Master-Master-Kommunikation verwendet.

Bei der Ankopplung als DP-Slave unterscheidet man nochmals 2 Varianten:

- Die Anbindung als Norm-Slave mittels einer GSD-Datei.
- Die Anbindung als intelligenter DP-Slave (i-Slave).

Als i-Slave werden Stationen bezeichnet, die eine eigenständige Intelligenz besitzen, und erst durch ihre eigene Programmierung in ihrem Funktionsumfang als DP-Slave festgelegt werden.

D.h., dass diese Stationen zuerst bzgl. Ihrer Kommunikationsstruktur fertig projektiert sein müssen, bevor sie als i-Slave weiterverwendet werden können.

Die verfügbaren i-Slaves finden Sie im HW-Katalog von HW Konfig im Ordner "bereits projektierte Stationen".

Unterschied: "Normaler" DP-Slave (Norm-Slave) - Intelligenter DP-Slave (I-Slave)

Bei einem "normalen" DP-Slave wie z. B. einem kompakten (ET 200eco) oder modularen (ET 200M) DP-Slave greift der DP-Master auf die dezentralen Ein-/Ausgänge zu.

Bei einem Intelligenten DP-Slave greift der DP-Master nicht auf die angeschlossenen Ein-/Ausgänge des Intelligenten DP-Slaves, sondern auf einen Übergabebereich im Ein-/Ausgangsadressraum der "vorverarbeitenden CPU" zu. Das Anwenderprogramm der vorverarbeitenden CPU muss für den Austausch der Daten zwischen Operandenbereich und Ein-/ Ausgängen sorgen.

Hinweis

Die projektierten E-/A-Bereiche für den Datenaustausch zwischen Master und Slaves dürfen nicht von E-/A-Baugruppen "besetzt" sein.

5.2.2 SIMOTION als DP-Slave an einer SIMATIC S7

5.2.2.1 Einleitung

Im Folgenden werden die Möglichkeiten beschrieben, wie ein SIMOTION Gerät als PROFIBUS-DP-Slave an ein PROFIBUS-Netzwerk angekoppelt werden kann.

Dazu gibt es 2 Möglichkeiten:

- Das SIMOTION Gerät wird als Norm-Slave mittels einer GSD-Datei an das DP-Mastersystem angebunden.
- Das SIMOTION Gerät wird als so genannter Intelligenter DP-Slave (i-Slave) in das DP-Mastersystem eingebunden

5.2.2.2 SIMOTION als DP-Slave mit Hilfe einer GSD-Datei an eine SIMATIC S7 koppeln

Vorgehensweise

Die GSD-Dateien für die verschiedenen SIMOTION Plattformen müssen zunächst in STEP7 HW Konfig importiert werden.

Die entsprechenden GSD-Dateien finden Sie auf der SIMOTION SCOUT CD "Add-on" im jeweiligen Geräteverzeichnis unter Firmware und Version.

Tabelle 5- 1 GSD-Datei

Gerät	Name der GSD Datei
SIMOTION C	Si0380aa.gsd
SIMOTION D4xx	Si0180ab.gsd (Diese Datei kann für alle SIMOTION D 4xx verwendet werden)
SIMOTION P	Si0280fa.gsd

Nachdem diese GSD-Dateien über das Menü Extras - GSD Datei installieren in STEP7 HW Konfig importiert wurden, erscheinen die Geräte im HW-Katalog unter weitere Feldgeräte - SPS - SIMATIC- SIMOTION und können von dort in ein DP-Mastersystem einer S7-Station eingefügt werden.

Hinweis

Auf SIMOTION Geräte, die per GSD-Datei an eine SIMATIC S7 angekoppelt wurden, kann nicht mit SIMOTION SCOUT über eine geroutete Verbindung zugegriffen werden.
Der Name einer GSC-Datei ist versionsabhängig, z.B. S10180AA und S10280AA.

Hinweis

Über einen Netzwerknoten hinweg kann auch auf Antriebe, welche als Einzelantrieb eingefügt wurden, geroutet werden.

Somit kann auf SIEMENS Antriebe, welche im SCOUT/STARTER projektiert werden können, auch geroutet werden, wenn diese als GSD-Slave / GSDML-Device in HW-Konfig projektiert werden. Hier gilt die Einschränkung, dass mittels Einstellung der Online-Zugangsparameter (**Zielgerät->Onlinezugang**) ein Netzübergangpunkt anhand der Subnetz ID eingestellt werden kann.

Ausserdem ist der Name der GSD-Datei versionsabhängig vergeben.

5.2.2.3 SIMOTION als i-Slave an eine SIMATIC S7 koppeln

Voraussetzung

- Auf dem Engineering-PC müssen SIMOTION SCOUT und damit STEP7 installiert sein.
- Die SIMATIC S7 und die SIMOTION Station müssen sich im gleichen Projekt befinden.

Sind diese Voraussetzungen gegeben, so kann die SIMOTION auch als so genannter i-Slave an das PROFIBUS DP Netz der SIMATIC angebunden werden.

Vorgehensweise

Es ist empfehlenswert, dass die SIMOTION Station als DP-Slave fertig projektiert ist, bevor sie als Slave am DP-Strang der SIMATIC CPU platziert wird.

Im folgenden ist die Vorgehensweise für eine SIMOTION C beschrieben. Die Vorgehensweise ist bis auf die Auswahl der SIMOTION Plattform gleich.

1. Konfigurieren einer Station als DP-Slave z. B. SIMOTION C-2xx
   Doppelklicken Sie auf die gewünschte Schnittstelle (z.B. DP2/MPI) in der Konfigurationstabelle und wählen Sie im Register Betriebsart die Option DP-Slave.
2. Konfigurieren der lokalen E/A-Adressen
   Im Register **Konfiguration** können Sie die lokalen E/A-Adressen und die Diagnoseadresse einstellen.
3. Wechseln Sie zu der projektierten SIMATIC Station, die DP-Master für die SIMOTION sein soll.
4. Anlegen eines i-Slave
   Ziehen Sie den Stationstyp "C2xx/P350/D4xx-i-Slave" aus dem Fenster Hardware Katalog (Ordner bereits projektierte Stationen) per Drag & Drop auf das Symbol für das DP-Mastersystem der SIMATIC Station.

5. Festlegen des intelligenten DP_Slave
   Doppelklicken Sie auf das Symbol für den Intelligenten DP-Slave SIMOTION und wählen das Register **Kopplung**. In diesem Register treffen Sie die Zuordnung, welche Station hier den intelligenten DP-Slave repräsentieren soll. In diesem Dialog finden Sie alle Stationen die bereits im Projekt vorhanden sind und als möglicher Koppelpartner in Betracht kommen.

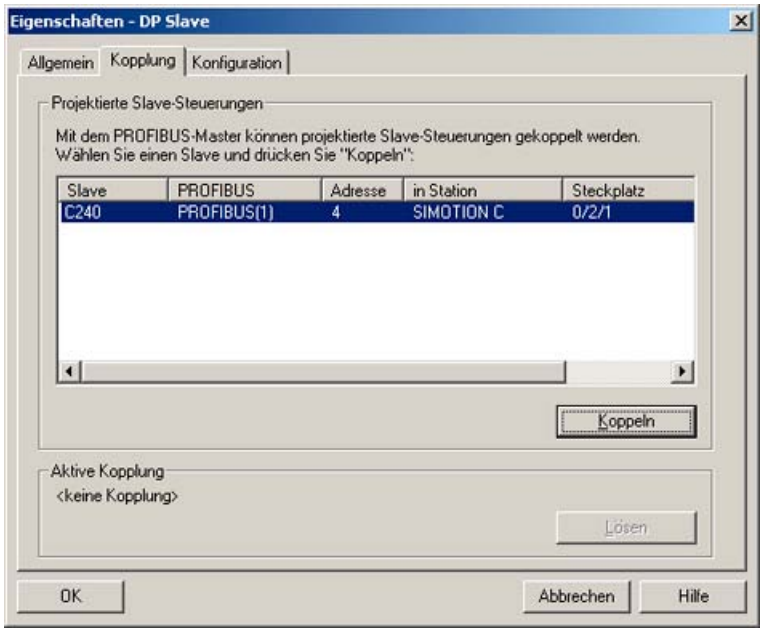


Bild 5-2 Eigenschaften DPSlave

6. Wählen Sie hier die entsprechende SIMOTION aus und drücken Sie auf **Koppeln**. Damit ist die projektierte SIMOTION Station nun als intelligenter DP-Slave an der SIMATIC angebunden

7. Wählen Sie das Register **Konfiguration** und ordnen Sie die Adressen einander zu:

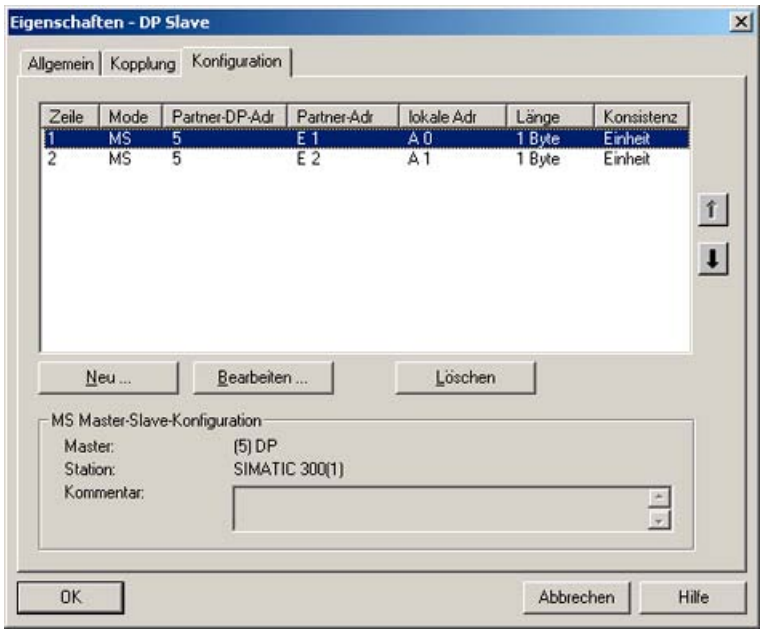


Bild 5-3 Eigenschaften - Konfiguration

- Für den Datenaustausch mit dem DP-Master über E/A-Bereiche wählen Sie den Mode **MS** (Master-Slave)
- Für den Direkten Datenaustausch mit einem DP-Slave oder DP-Master wählen Sie den Mode **DX** (Direct Data Exchange)

1. Bestätigen Sie die Einstellungen mit **OK**.

Damit ist die Projektierung der SIMOTION Station als intelligenter DP-Slave an der SIMATIC Station abgeschlossen und die Daten können über die angegebenen E/A-Adressen ausgetauscht werden.

5.2.3 SIMATIC S7 als DP-Slave an einer SIMOTION

5.2.3.1 Einleitung

Im Folgenden werden die Möglichkeiten beschrieben, wie eine SIMATIC Station als PROFIBUS-DP-Slave an einen PROFIBUS-Netzwerk angekoppelt werden kann.

Dazu gibt es 2 Möglichkeiten:

- Die SIMATIC-Station wird als Normslave mittels einer GSD-Datei an das DP-Mastersystem einer SIMOTION angebunden.
- Die SIMATIC-Station wird als so genannter i-Slave in das DP-Mastersystem einer SIMOTION eingebunden.

5.2.3.2 SIMATIC als DP-Slave mit Hilfe einer GSD-Datei an ein SIMOTION Gerät koppeln

Vorgang

Die GSD-Dateien für die verschiedenen SIMATIC-Stationen müssen zunächst in STEP7 HW Konfig importiert werden.

Die entsprechenden GSD-Dateien finden Sie im Produktsupport unter: http://support.automation.siemens.com/ww/view/de/113653.

Nachdem diese GSD-Dateien über das Menü Extras - GSD Datei installieren in STEP7 HW Konfig importiert wurden, erscheinen die Geräte im HW-Katalog unter weitere Feldgeräte - SPS - SIMATIC und können von dort in ein DP-Mastersystem einer SIMOTION Station eingefügt werden.

Auf SIMATIC S7 Geräte, die per GSD-Datei an ein SIMOTION Gerät angekoppelt wurden, kann nicht mit STEP7 über eine geroutete Verbindung zugegriffen werden.

5.2.3.3 SIMATIC S7 CPU als i-Slave an ein SIMOTION Gerät koppeln

Voraussetzungen

- Auf dem Engineering-PC ist SIMOTION SCOUT und damit auch SIMATIC STEP7 installiert.
- Die SIMATIC S7- und die SIMOTION Station müssen sich im gleichen Projekt befinden

Sind diese Voraussetzungen gegeben, so kann die SIMATIC auch als so genannter i-Slave an das PROFIBUS-DP Netz der SIMOTION angebunden werden.

Vorgehensweise

Es ist empfehlenswert, dass die SIMATIC Station als DP-Slave fertig projektiert ist, bevor sie als Slave am DP-Strang der SIMOTION platziert wird.

Im folgenden ist die Vorgehensweise für eine CPU 315-2 DP beschrieben. Die Vorgehensweise ist bis auf die Auswahl der CPU-Typen auch bei einer S7-400 gleich.

1. Konfigurieren Sie eine Station z.B. mit der CPU 315-2 DP als DP-Slave. Doppelklicken Sie auf die Zeile 2.1 (Schnittstelle) in der Konfigurationstabelle und wählen Sie die Option DP-Slave im Register **Betriebsart**.
2. Im Register **Konfiguration** können Sie die lokalen E/A-Adressen und die Diagnoseadresse einstellen
3. Wechseln Sie zu der projektierten SIMOTION Station, die DP-Master für die SIMATIC sein soll.
4. Ziehen Sie den entsprechenden Stationstyp CPU 31x bzw. CPU 41x aus dem Fenster Hardware Katalog (Ordner bereits projektierte Stationen) per Drag & Drop auf das Symbol für das DP-Mastersystem der SIMOTION Station.

5. Doppelklicken Sie auf das Symbol für den Intelligenten DP-Slave und wählen das Register Kopplung. In diesem Register treffen Sie die Zuordnung, welche Station hier den Intelligenten DP-Slave repräsentieren soll. In diesem Dialog finden Sie alle Stationen die bereits im Projekt vorhanden sind und als möglicher Koppelpartner in Betracht kommen.

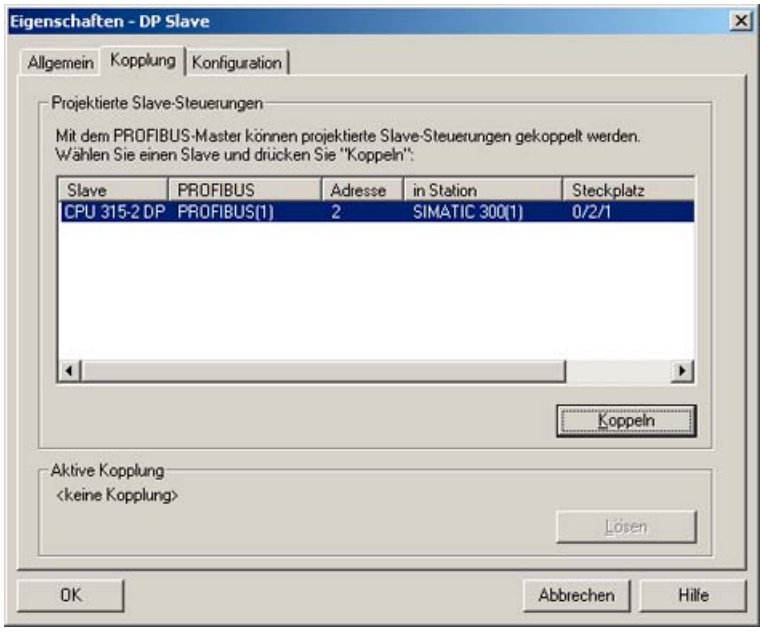


Bild 5-4 Eigenschaften - Kopplung

6. Wählen Sie hier die entsprechende S7-Station aus und drücken Sie auf **Koppeln**. Damit ist die projektierte S7-Station nun als intelligenter DP-Slave an der SIMOTION angebunden.

7. Wählen Sie das Register **Konfiguration** und ordnen Sie die Adressen einander zu:

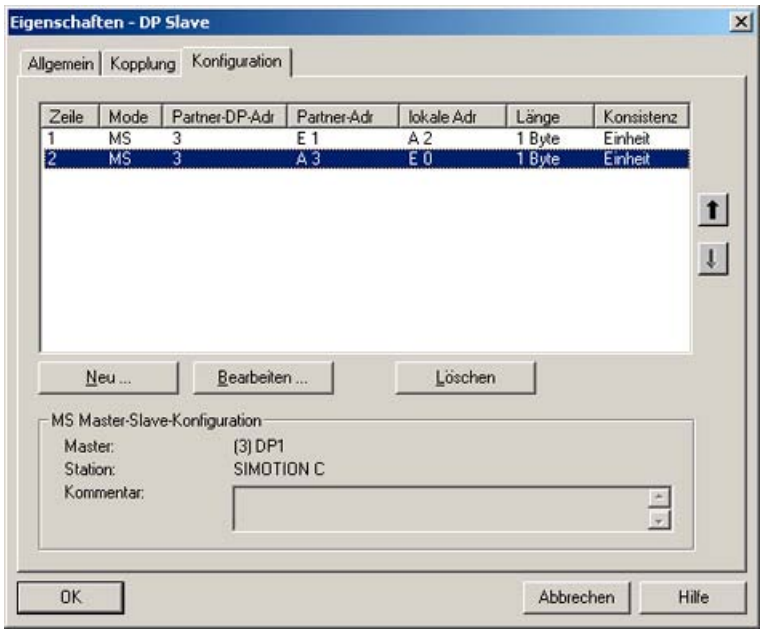


Bild 5-5 Konfiguration - Adressauswahl

   – Für den Datenaustausch mit dem DP-Master über E/A-Bereiche wählen Sie den Mode **MS** (Master-Slave)
   – Für den Direkten Datenaustausch mit einem DP-Slave oder DP-Master wählen Sie den Mode **DX** (Direct Data Exchange)

8. Bestätigen Sie die Einstellungen mit **OK**.

Damit ist die Projektierung der SIMATIC-Station als intelligenter DP-Slave an der SIMOTION Station abgeschlossen und die Daten können über die angegebenen E/A-Adressen ausgetauscht werden.

5.2.4 PROFIBUS-Master-Master-Verbindung zwischen SIMATIC und SIMOTION

5.2.4.1 Einleitung

Master-Master-Kommunikation

Eine Master-Master-Kommunikations-Verbindung zwischen einem SIMATIC S7 und einem SIMOTION Gerät über PROFIBUS wird durch den Einsatz der Systemfunktionen SFC65 (XSEND) und SFC66 (XRECEIVE) auf SIMATIC-Seite und den Systemfunktionen _Xsend und _Xreceive auf SIMOTION-Seite erstellt. Eine Projektierung der Kommunikations-Verbindung in NetPro ist nicht erforderlich.

Tabelle 5- 2 Master-Master Kommunikation

Protokoll	SIMATIC Gerät	Funktion	SIMOTION Gerät	Funktion
PROFIBUS	S7-300 CPU S7-400 CPU	SFC65 (XSEND) SFC66 (XRCV)	C2xx D4xx P350	_Xsend _Xreceive

Die Vergabe der PROFIBUS-Adressen erfolgt in HW Konfig. Alle weiteren Bausteinparameter werden vom Anwender für die Verbindung festgelegt und beim Aufruf der Funktion mit übergeben. Somit verhält es sich mit der PROFIBUS-Verbindung zwischen SIMATIC und SIMOTION ähnlich wie mit einer TCP-/IP-Verbindung zwischen einer SIMATIC-Station mit integrierter Ethernetschnittstelle und einem SIMOTION Gerät bzw. umgekehrt. Die für die Kommunikation wichtigen Parameter werden vom Anwender bestimmt und beim Baustein- bzw. Funktionsaufruf übergeben.

Im folgenden Kapitel wird auf die Parametrierung der Systemfunktionen auf SIMATIC S7-Seite und der Funktionen auf SIMOTION-Seite näher eingegangen

5.2.4.2 SIMATIC S7-Systemfunktionen für eine PROFIBUS-Verbindung

Einleitung

Die PROFIBUS-Verbindung zwischen einer SIMATIC S7 Station und einem SIMOTION-Gerät wurde im vorhergehenden Kapitel vorgestellt. Im Folgenden soll nun die Parametrierung der SIMATIC S7-Systemfunktionen bzw. der SIMOTION-Funktionen für eine PROFIBUS-Verbindung näher erläutert werden.

SIMATIC S7-Systemfunktionen

Zur Kommunikation zwischen einer SIMATIC S7 Station und einem SIMOTION-Gerät werden auf SIMATIC S7-Seite die beiden Systemfunktionen SFC65 X_SEND und SFC66 X_RCV verwendet.
SIMOTION Funktionen:

```
CALL "X_SEND"
    REQ:=M1.0
    CONT:=FALSE            //Dies ist die DP-Adresse des
    DEST_ID:=W#16#2        //Kommunikationspartners (SIMOTION P350)
    REQ_ID:=DW#16#2        //Die REQ_ID muss mit der MessageID auf
    SD:=P#DB100.0DBX0.0 BYTE 10 //der SIMOTION-Empfang-Seite
    //übereinstimmen!
    RET_VAL:=MW64
    BUSY:=M1.1
```

Parametrierung der Systemfunktion SFC65 X_SEND

Um Daten über eine PROFIBUS-Verbindung von einer SIMATIC S7 Station an ein SIMOTION-Gerät zu versenden, wird auf SIMATIC S7-Seite die Systemfunktion SFC65 X_SEND aufgerufen.

Über den Parameter REQ wird die Datenübertragung gesteuert. d.h. wird der Parameter auf den Wert 1 gesetzt, so wird die Datenübertragung gestartet. Falls zu diesem Zeitpunkt noch keine Verbindung zum Kommunikationspartner besteht, wird sie vor dem Senden der Daten aufgebaut.

Der Parameter CONT dient der Parametrierung des Verhaltens der Verbindung nach Beendigung der Datenübertragung. Wird in den Parameter CONT der Wert 1 eingetragen, so wird die Verbindung nach Beendigung der Datenübertragung gehalten. Wird als Wert 0 eingetragen, so wird nach erfolgter Datenübertragung die Verbindung abgebaut.

Der Parameter DEST_ID enthält die PROFIBUS-Adresse des SIMOTION-Gerätes. Sie wird in STEP 7 HW Konfig festgelegt.

REQ_ID kennzeichnet die Sendedaten. d.h. über den Wert im Parameter REQ_ID können die gesendeten Daten im SIMOTION-Gerät eindeutig der S7-Station zugeordnet werden. Der hier vergebene Wert wird im Parameter messageid in der Empfangsfunktion auf SIMOTION-Seite.zurück gemeldet.

Mit SD wird der Bereich festgelegt, aus dem die Sendedaten stammen.

Die beiden Parameter RET_VAL und BUSY dienen der Beobachtung des Status des Sendevorgangs. BUSY zeigt an, ob der Sendeauftrag noch läuft oder schon vollständig ausgeführt wurde. Über RET_VAL ist insbesondere im Fehlerfall eine detailliertere Diagnose möglich.

```
CALL "X_RCV"
    EN_DT:=M0.0
    RET_VAL:=MW50
    REQ_ID:=MD52
    NDA:=M0.1
    RD:=P#DB110.DBX0.0 BYTE 10
```

Aufruf-Beispiel der Systemfunktion SFC66 X_RCV

Sollen auf einer SIMATIC S7 Station Daten von einem SIMOTION-Gerät empfangen werden, so ist im S7-Programm die Systemfunktion SFC66 X_RCV aufzurufen.

Am Eingang "EN_DT" der Systemfunktion wird angegeben:

- ob die Funktion nur auf das Eintreffen neuer Daten überprüfen soll (EN_DT=0) oder
- ob die empfangenen Daten nach Empfang aus der Warteschlange in den durch "RD" vorgegebenen Bereich umkopiert werden sollen (EN_DT=1).

Mit dem Parameter RET_VAL kann der Anwender den Status des Funktionsaufrufs beobachten. Speziell im Fehlerfall erhält der Anwender ausführlichere Informationen zur Ursache des Fehlers.

REQ_ID kennzeichnet die Empfangsdaten, d.h. über den Parameter REQ_ID können die empfangenen Daten einem SIMOTION-Gerät eindeutig zugeordnet werden. Der hier empfangene Wert entspricht dem Wert im Parameter messageid in der entsprechenden Sendefunktion auf SIMOTION-Seite.

Der Parameter NDA zeigt an, ob neue Daten empfangen wurden. Wenn NDA auf 1 steht, so stehen neue Daten zur Verfügung und können in den Empfangsdatenbereich übertragen werden. Hat NDA den Wert 0, so sind keine neuen Daten vorhanden.

Mit dem Parameter RD wird angegeben, wo die empfangenen Daten abgelegt werden.

SIMOTION-Funktionen

```
RetVal_PB_Senden:=
    _xsend(PB_Senden_CommunicationMode, PB_Senden_Address,
    PB_Senden_MessageID, PB_Sender_NextCommand, PB_Senden_CommandID,
    PB_Sende_Daten, PB_Sende_Daten_Laenge);
```

Beispiel für den Aufruf der SIMOTION-Funktion_xsend

Kommunizieren die SIMATIC S7 Station und das SIMOTION-Gerät über PROFIBUS, so wird auf SIMOTION-Seite zum Senden die Funktion _xsend aufgerufen.

Mit dem Parameter "communicationmode" wird der aufgerufenen Funktion mitgeteilt, was mit der Verbindung nach erfolgreicher Datenübertragung passieren soll. Der Datentyp der Funktion sieht die Vergabe der Werte ABORT_CONNECTION oder HOLD_CONNECTION vor. Wird der Parameter mit ABORT_CONNECTION vorbelegt, so wird die Verbindung nach der Datenübertragung abgebaut. Mit dem Wert HOLD_CONNECTION wird die Funktion so parametriert, dass nach erfolgreicher Datenübertragung die Verbindung bestehen bleibt.

Hinter dem Parameter address verbirgt sich eine Struktur vom Datentyp StructXsendDestAddr, die wiederum aus verschiedenen Parametern zusammengesetzt ist. Diese Struktur enthält alle Informationen zur Adresse des Kommunikationspartners des SIMOTION-Gerätes.

Parameter-Struktur "StructXsendDestAddr

Im Folgenden sollen die einzelnen Parameter der Struktur aufgeführt und erläutert werden.

Mit dem Parameter deviceid wird der jeweiligen SIMOTION-Hardware Rechnung getragen. Mit dem Parameter wird der physikalische Anschlusspunkt der Verbindung angegeben. So wird für eine SIMOTION C2xx für die Schnittstelle X8 der Wert 1 eingetragen. Für die Schnittstelle X9 wird der Wert 2 angegeben. Ist die SIMATIC S7 Station an X101 einer SIMOTION P350 angeschlossen, so wird in den Parameter deviceid der Wert 1 vergeben. Für die Schnittstelle X102 wird in den Parameter deviceid der Wert 2 geschrieben. Bei der

SIMOTION D4x5 wird für die Schnittstelle X126 der Wert 1 und für X136 der Wert 2 in den Parameter deviceid eingetragen.

Da für die Kommunikation über MPI oder PROFIBUS keine Subnetzmaske angegeben wird, wird der Paramter remotesubnetidlength mit dem Wert 0 vorbelegt. Als Konsequenz ergibt sich, dass die Belegung des Parameters remotesubnetid irrelevant ist.

Der Parameter remotestaddrlength wird für die MPI- bzw. PROFIBUS-Kommunikation mit dem Wert 1 belegt.

Mit dem Parameter nextstaddrlength wird die Länge der Router-Adresse angegeben. Da für die MPI- oder PROFIBUS-Kommunikation zwischen der SIMATIC S7 Station und dem SIMOTION-Gerät keine Router eingesetzt werden, wird für diesen Parameter der Wert 0 angegeben. Folglich ist auch der Parameter nextstaddr ohne Relevanz (siehe auch unten).

Der folgende Parameter remotesubnetid bezeichnet die Subnetzmaske und hat, wie bereits oben erwähnt, für die Kommunikation über MPI oder PROFIBUS keine Bedeutung.

Mit dem Parameter remotestaddr wird die eigentliche Zieladresse angegeben. Bei dem Parameter handelt es sich um ein Array. Allerdings wird für die MPI- oder PROFIBUS-Kommunikation nur der erste Index verwendet. Die weiteren fünf Indizes haben keinerlei Bedeutung.

Der Parameter nextstaddr dient der Angabe der Router-Adresse. Für ihn gilt dasselbe wie für den Parameter remotesubnetid. Auch seine Belegung hat für die Kommunikation über MPI oder PROFIBUS keine Relevanz.

Zur Identifizierung des SIMOTION-Gerätes auf der Empfangsseite wird der Parameter messageid durch den Anwender belegt. Der eingetragene Wert ermöglicht auf der SIMATIC S7 Station eine Zuordnung über den Parameter REQ_ID. Dort kann der Wert aus dem Parameter messageid ausgelesen werden.

Für diese Funktion wird mit dem Parameter nextcommand das Verhalten der Funktion in Bezug auf das Weiterschalten beim Aufruf parametriert. Es gibt zwei Einstellmöglichkeiten: IMMEDIATELY und WHEN_COMMAND_DONE. Beim ersten Wert wird unmittelbar oder beim zweiten Wert erst nach Beendigung des Befehls weitergeschaltet.

Beim Aufruf der Funktion wird ihr im Parameter commandid eine systemweit eindeutige Nummer vergeben, um den Befehlsstatus verfolgen zu können.

Die Sendedaten werden über die Variable data der Funktion beim Aufruf übergeben.

Die Länge der aus dem Sendebereich zu übertragenden Daten wird über den Parameter datalength angegeben.

Der Rückgabewert der Funktion _xsend an das Anwenderprogramm hat den Datentyp DINT und über die unterschiedlichen Rückgabewerte wird auf Probleme bei der Ausführung der Funktion hingewiesen. Ebenso wird zurückgemeldet, wenn die Daten erfolgreich gesendet wurden.

```
RetVal_PB_Empfanen:=
    _xreceive(PB_Empfangen_MessageID,
    PB_Empfangen_NextCommand,PB_Empfangen_CommandID);
```

Aufruf-Beispiel der SIMOTION-Funktion_xreceive

Das Beispiel zeigt die Verwendung der Funktion _xreceive. Die Funktion wird verwendet, wenn Daten von SIMATIC S7 Station über PROFIBUS empfangen werden sollen.

Zur Identifizierung der S7 Station, von der die Daten empfangen werden sollen, wird der Funktion _xreceive der Parameter messageid übergeben. Es wird der Wert eingetragen, der auf der S7-Seite im Parameter REQ_ID der korrespondierenden Systemfunktion _xsend vergeben wurde.

Für diese Funktion wird mit dem Parameter nextcommand das Verhalten der Funktion in Bezug auf das Weiterschalten beim Aufruf parametriert. Es gibt zwei Einstellmöglichkeiten: IMMEDIATELY und WHEN_COMMAND_DONE. Beim ersten Wert wird unmittelbar oder beim zweiten Wert erst nach Beendigung des Befehls weitergeschaltet.

Beim Funktionsaufruf wird im Parameter commandid eine systemweit eindeutige Nummer vergeben, um den Befehlsstatus verfolgen zu können.

Die Struktur, die von der Funktion ans Anwenderprogramm zurückgeliefert wird, beinhaltet die Parameter functionresult, datalength und data. Über den Parameter functionresult kann der Status des Empfangens abgefragt werden. Der Parameter datalength meldet die Anzahl der empfangenen Nutzdatenbytes nach erfolgreichem Aufruf der Funktion _xreceive zurück. Über den Parameter data kann auf die empfangenen Nutzdaten zugegriffen werden.

Ethernet Allgemein (TCP/IP- und UDP-Verbindungen)

6

6.1 Einleitung

Nachfolgend wird beschrieben, wie offene TCP-/IP- und UDP-Ethernet-Verbindungen zwischen einem SIMOTION Gerät und einem SIMATIC S7 Gerät ausgeführt werden können.

Es werden alle notwendigen vorzubereitenden Schritte und anhand einer beispielhaften Programmierung die erforderlichen Funktionsaufrufe erläutert.

6.2 Ethernet-Subnetze projektieren mit SIMOTION

6.2.1 Eigenschaften der Ethernet Subnetze

SIMOTION besitzt je nach Gerät eine oder zwei Onboard Ethernet-Schnittstellen. An die 8-poligen RJ45-Buchsen können Sie ein Industrial Ethernet mit einer Übertragungsgeschwindigkeit von 10/100 MBit/s anschließen.

Alternativ können Sie eine Industrial Ethernet auch über die PROFINET-Baugruppen, wie z.B. CBE30 einer SIMOTION D4x5, anschließen.

Über einen PG/PC können Sie mit STEP 7, SIMOTION SCOUT und SIMATIC NET OPC kommunizieren.

Über TCP/IP kann auch mit anderen Geräten wie SIMOTION Geräte, SIMATIC S7 Geräte oder PC kommuniziert werden.

Bei Baugruppen mit 2 Ethernet-Schnittstellen gibt es keine HUB/Switch-Funktionalität, d.h. Telegramme werden nicht von einer Schnittstelle zur anderen weitergeleitet. Die Schnittstellen gehören zu getrennten Ethernet-Subnetzen. Die SIMOTION Geräte haben keine IP-Routerfunktionalität, sie leiten die Telegramme nicht von einem Subnetz zum anderen.

TCP/IP-Timeout-Parameter sind bei 2 Schnittstellen einmal für beide Schnittstellen einstellbar. Die Übertragungsrate/Duplex sind bei 2 Schnittstellen für beide Schnittstellen getrennt einstellbar.

Es werden "Dienste über TCP" unterstützt für beide Ethernet-Schnittstellen, darüber ist ein S7-Routing von den Ethernet-Schnittstellen in die Profibus-Schnittstellen hinein möglich. Die "Dienste über TCP/IP" werden nicht von der einen Ethernet-Schnittstelle in die andere geroutet.

Die MAC-Adressen sind außen auf dem Gehäuse sichtbar.

Verwendung

Industrial Ethernet können Sie mit SIMOTION wie folgt verwenden:

- zur Kommunikation mit STEP7, SIMOTION SCOUT und SIMATIC NET OPC über ein PG/PC
- zur Kommunikation über UDP (User Datagram Protocol) mit anderen Komponenten, z.B. anderen SIMOTION Geräten, SIMATIC Geräten oder PCs
- zur Kommunikation über TCP (Transfer Control Protocol) mit anderen Komponenten, z.B. anderen SIMOTION Geräten, SIMATIC S7 Stationen oder PC
- zum Anschließen von SIMATIC HMI Geräten wie z.B. MP277, MP370 oder PC-basierte HMIs
- zur Kommunikation mittels SIMOTION IT DIAG und SIMOTION IT OPC XML-DA (jeweils eigene Lizenz erforderlich)
- zur Kommunikation mittels SIMOTION VM (eigene Lizenz erforderlich)

6.3 Funktionsübersicht und Funktionsablauf einer Ethernet Kommunikation über TCP/IP oder UDP

6.3.1 Einleitung

Nachfolgend wird beschrieben, welche System- bzw. Kommunikationsfunktionen für eine projektierte Ethernet-Kommunikationsverbindung zur Verfügung stehen, und wie diese Funktionen in der richtigen Ablauf-Sequenz eingesetzt werden.

6.3.2 SIMOTION TCP/IP Funktionen - Modellierung

Die Kommunikations-Sequenz ist in der Abbildung **Prinzipieller Ablauf der Kommunikations-Sequenz einer TCP/IP Kommunikation** anhand der SIMOTION Systemfunktionen dargestellt und erläutert.

Bei Kommunikationsverbindungen mit einer SIMATIC S7 müssen diese Systemfunktionen durch die entsprechenden S7-Systemfunktionsbausteine übernommen werden. Eine entsprechende Gegenüberstellungstabelle finden Sie unter **SIMATIC Funktionen**.

Die Modellierung erläutert die in der Sequenz aufgezeigten Einzelschritte

- Server setzt sich an Port wartend (1)
- Client meldet Verbindungsanforderung auf diesen Port an. (2). Ist serverseitig kein Port angemeldet, wird mit TimeOut (Systemeinstellung) gewartet
- Server legt bei Verbindungsanmeldung internen Kommunikationsport an und gibt den Serverport für neuen Verbindungsaufbau frei. Der interne Kommunikationsport wird über die connectionId identifiziert (3)
- Daten senden / empfangen, sowohl vom Client wie auch vom Server über diese Verbindung möglich (4)

- am Serverport können weitere Verbindungen aufgebaut werden (5)
- eine bestehende Verbindung kann client- oder serverseitig mit _tcpCloseConnection geschlossen werden (6)
- Serverport für Verbindungsaufbau wird mit _tcpCloseServer geschlossen (7)

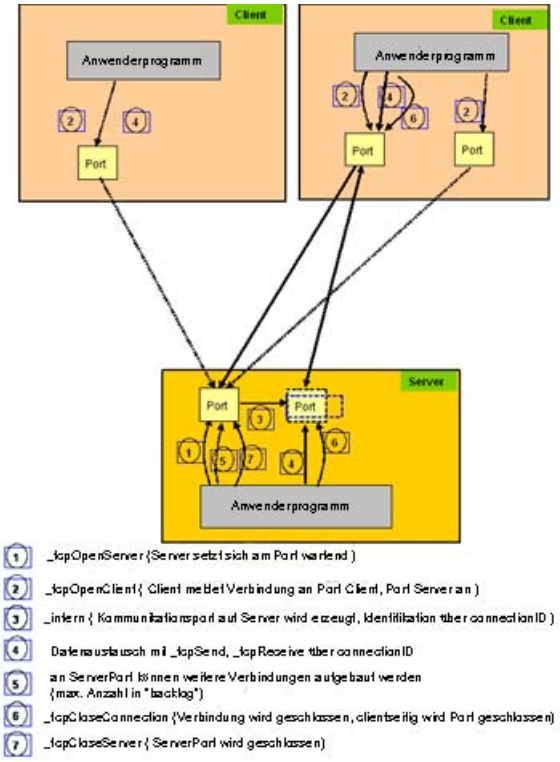


Bild 6-1 Prinzipieller Ablauf der Kommunikations-Sequenz einer TCP/IP Kommunikation

6.3.3 SIMOTION TCP/IP-Funktionen - Erläuterung

Die 3 Aufrufe von _tcpOpenClient in der Modellierung beziehen sich alle auf den gleichen Server (IP-Adresse/Port). Intern wird auf dem Server dafür allerdings jeweils ein eigener Port zugewiesen. Nach außen wird die Kommunikation mit der connectionId durchgeführt.

Portvergabe:

- Die Portnummer liegt im Bereich 1024 bis 65535.
- Der Port am Client kann gleich dem Port am Server sein.
- Der Port am Client kann ungleich dem Port am Server sein.

Die Sequenz zeigt ein einfaches Beispiel für die Funktionsablaufreihenfolge bei 2 Partnern:

Tabelle 6- 1 Kommunikation zwischen einem Sender (Client) und einem Empfänger (Server)

	Funktion
Kommunikationsverbindung aufbauen • Empfänger/Server wartet auf Kommunikationsverbindung • Sender/Client fordert einen Verbindungsaufbau zum Empfänger • Empfänger/Server hat Verbindungsanforderung aufgebaut • Es wird keine weitere Verbindung mehr benötigt	 _tcpOpenServer _tcpOpenClient _tcpCloseServer
Kommunizieren • Sender sendet Daten an den Empfänger • Empfänger empfängt Daten vom Sender	 _tcpSend _tcpReceive
Kommunikationsverbindung beenden • Sender sendet keine Daten mehr und schließt die Verbindung	 _tcpCloseConnection

Ein Sender oder Empfänger kann beim Verbindungsaufbau sowohl Client als auch Server sein. Beim TCP/IP Verbindungsaufbau muss es mindestens einen Client und einen Server geben.

Die Client-Server-Beziehung gilt nur bis zum Abschluss des Verbindungsaufbaus. Nach dem Verbindungsaufbau sind beide Kommunikationspartner gleichwertig, d.h. jeder der beiden kann zu einem beliebigen Zeitpunkt senden oder empfangen oder die Verbindung schließen.

6.3.4 SIMOTION UDP-Funktionen - Modellierung

Beschreibung des UDP (User Datagram Protocol)

UDP (User Datagram Protocol) stellt ein Verfahren zur Verfügung, um aus dem Anwenderprogramm mit einem Minimum von Protokoll Mechanismus Daten über Ethernet zu senden und zu empfangen. Eine Rückinformation bezüglich der übertragenen Daten findet bei Kommunikation über UDP nicht statt.

Die Kommunikation findet sendeseitig und empfangsseitig über Ports statt.

Im Gegensatz zu TCP/IP müssen Sie keinen Verbindungsaufbau bzw. -abbau programmieren.

UDP-Kommunikationsmodell

- Für den Empfang adressieren Sie im Befehl den Port, den Sie auf Ihrer Komponente für den Kommunikationsauftrag nutzen möchten.
- Sie geben beim Senden von Daten die IP-Adresse des Zielsystems, die dortige Port-Nummer für die Daten und die Port-Nummer Ihrer Komponente (siehe oben) an.
- Sie können angeben, ob der Port auf Ihrer Seite nach der Ausführung des Kommunikationsauftrages reserviert bleiben soll.

- UDP ist kein gesichertes Modell. Es können deshalb Daten bei der Übertragung verloren gehen. Eine gesicherte Datenübertragung müssen Sie über Ihre Applikation programmieren, z.B. durch Quittieren des Datenempfangs.
- Mit der Funktion _udpReceive können Sie die Daten eines Übertragungsprotokolls in die Rückgabestruktur übertragen, sind mehrere Datenprotokolle mit _udpReceive zurückgegeben worden, wird das "älteste" Datenprotokoll zurückgegeben.

Folgendes Bild zeigt das UDP-Kommunikationsmodell auf SIMOTION-Seite

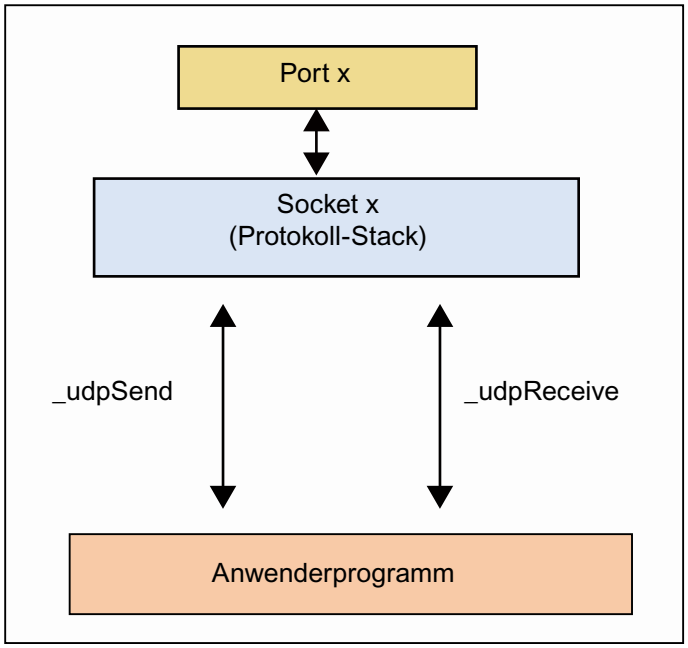


Bild 6-2 UDP-Kommunikationsmodell

Siehe auch

Funktion _udpSend (Seite 134)

Funktion _udpReceive (Seite 135)

UDP-Verbindung (Seite 109)

6.3.5 SIMATIC Funktionen

Nachfolgend finden Sie die Gegenüberstellung von den SIMOTION Systemfunktionen zu den benötigten, korrespondierenden SIMATIC S7-Funktionen, um eine Ethernet-Kommunikation zwischen einer SIMOTION und einer SIMATIC S7 zu etablieren.

In der nachfolgenden Tabelle ist zunächst die Zuordnung der Kommunikations-Funktionen zu den Protokollen und einzelnen Geräten angegeben.

Tabelle 6- 2 Übersicht der Protokolle, Geräte und Kommunikations-Funktionen

Protokoll	SIMATIC Gerät	Funktion	SIMOTION Gerät	Funktion
TCP/IP	S7 300-CPU mit Ethernet-CP (CP343-1)	FC5 AG_SEND, FC6 AG_RECV	C2xx D4xx P350	_tcpOpenClient, _tcpSend, _tcpReceive, _tcpCloseConnection, _tcpOpenServer, _tcpCloseServer
	S7 300-CPU mit integrierter Ethernet-Schnittstelle	FB63 TSEND, FB64 TRCV, FB65 TCON, FB66 TDISCON, UDT65 TCON_PAR		
	S7 400-CPU	FC50 AG_LSEND, FC60 AG_LRECEIVE		
UDP	S7 300-CPU mit Ethernet-CP (CP343-1)	FC5 AG_SEND, FC6 AG_RECEIVE	C2xx D4xx P350	_udpSend, _udpReceive
	S7 300-CPU mit integrierter Ethernet-Schnittstelle	Dieses Protokoll wird von CPU-Baugruppen der Reihe S7 300 mit integrierter Ethernet-Schnittstelle nicht unterstützt!		
	S7 400-CPU	FC50 AG_LSEND, FC60 AG_LRECEIVE		
PROFIBUS	S7 300-/S7 400-CPU	SFC65 (XSEND) SFC66 (XRCV)	C2xx D4xx P350	_Xsend _Xreceive

Für S7-300 mit Ethernet-CP gilt:

Bei den akutellen Ausgabeständen der Ethernet-CP werden ausschließlich die FC AG_SEND/AG_RECV verwendet; die Datenlänge kann hier bis zu 8192 Byte (siehe Tabelle) betragen. Bei älteren Ausgabeständen der Ethernet-CP ist die Datenlänge pro Auftrag auf <=240 Byte beschränkt (gilt bis Bausteinversion V3.0 von AG_SEND/AG_RECV); bei späteren Ausgabeständen der Ethernet-CP können auch längere Daten (bis zu 8192 Byte) mit Hilfe der FC AG_LSEND oder AG_LRECV übertragen werden.

D.h. es ist wichtig den Ausgabestand des CP und den Ausgabestand der verwendeten Bausteine zu kennen und zu beachten.

Für S7-400 gilt:

Für die S7-400 können ebenfalls die FC AG_SEND/AG_RECV verwendet werden. Allerdings ist hier die übertragbare Datenlänge generell pro Auftrag auf <=240 Byte beschränkt!

Längere Datensätze (maximal 8192 Byte; siehe Tabelle) können mit Hilfe der FC AG_LSEND/AG_LRECV übertragen werden. Auch hier ist es wichtig zu wissen, welche Datenlänge der verwendete CP unterstützt. Dies kann der Beschreibung des CP entnommen werden.

Die Tabelle gibt einen Überblick über die Datenmengen die mit den verschiedenen Übertragungsverfahren zwischen SIMATIC S7 und SIMOTION ausgetauscht werden können.

Tabelle 6- 3 Maximale übertragbare Länge je Auftrag mit den Kommunikationsfunktionen

Funktion	TCP-/IP-Protokoll	UDP-Protokoll	PROFIBUS-Protokoll
FC5 AG_SEND, FC6 AG_RECV (S7-300)	8192 Byte	2048 Byte	
FC50 AG_LSEND, FC60 AG_LRECV (S7-400)	8192 Byte	2048 Byte	
FB63 TSEND, FB64 TRCV (S7-300 CPU mit integrierter Ethernet-Schnittstelle	1460 Byte	Dieses Protokoll wird von CPU-Baugruppen der Reihe S7-300 mit integrierter Ethernet-Schnittstelle nicht unterstützt!	
_tcpSend, _tcpReceive, udpSend, udpReceive (SIMOTION C2xx, D4xx, P350	4096 Byte	1470 Byte	
SFC65 (XSEND), SFC66 (XRCV) (S7-300, S7-400)			76 Byte
_Xsend, _Xreceive (SIMOTION C2xx, D4xx, P350)			200 Byte

6.3.6 Allgemeine Hinweise

Die Kommunikation über Ethernet ist verbindungsorientiert, d.h. erst wenn eine Verbindung zur Partnerstation aufgebaut ist, können Daten übertragen werden.

Die TCP/IP Kommunikation geschieht über Datenpakete, die vom Sender in einer bestimmten Größe gesendet werden. Diese können jedoch beim Empfänger in unterschiedlichen Datenpaketgrößen ankommen.

Auf Empfängerseite sind folgenden Szenarien möglich:

- Teilpakete:
  empfangenes Datenpaket < gesendetes Datenpaket
- Mehrere Pakete zu einem großen Datenpaket zusammengefasst:
  empfangenes Datenpaket > gesendetes Datenpaket

Die Reihenfolge der Daten bleibt dabei erhalten. Der Anwender muss in seinem SIMOTION Programm dafür sorgen, dass diese Datenpakete wieder zu der Länge des gesendeten Datenpakets zusammengebaut werden. Details erfahren Sie in den entsprechenden Projektierungskapiteln.

Die Ethernet Kommunikation von SIMOTION mit TCP/IP bzw. UDP ist möglich.

- mit einer SIMATIC S7 Baugruppe mit Ethernetanschluss (integriert oder mit extra Ethernet CP). Welche SIMATIC S7-Baugruppe TCP/IP beherrscht, ist den technischen Angaben zu der jeweiligen Baugruppe zu entnehmen. Die wesentlichen Baugruppentypen sind in der Tabelle **Übersicht der Protokolle, Geräte und Kommunikations-Funktionen** unter SIMATIC Funktionen (Seite 99) angegeben
- mit einem PC. Dazu ist auf dem PC eine entsprechende Software nötig, die TCP/IP-Kommunikation unterstützt (z. B. Perl, Visual Basic oder C++).
- zwischen den SIMOTION Baugruppen Cxx, Dxx und P350.

Die TCP/IP System-Funktionen der SIMOTION dürfen nur in der BackgroundTask oder in einer MotionTask aufgerufen werden.

Nutzdaten im Stack beim Datenaustausch

TCP und UDP setzen auf das IP-Protokoll auf. Bei der Kommunikation über TCP bzw. UDP werden die empfangen Nutzdaten im Stack der SIMOTION-CPU vorgehalten. Dort müssen die Nutzdaten per Applikation abgeholt werden. Die Größe der vorgehaltenen Nutzdaten ist pro Verbindung begrenzt:

- UDP-Kommunikation 1470 Bytes Nutzdaten
  (Die Nutzdatengröße pro Telegramm ist durch die Systemfunktion begrenzt:
  < V4.1 SP4 1400 Bytes
  ab V4.1 SP4 1470 Bytes).
- TCP/IP-Kommunikation 8192 Bytes Nutzdaten

Hinweis

Bei der UDP-Kommunikation muss die Applikation daher dafür sorgen, dass die Nutzdaten rechtzeitig abgeholt werden, bevor der Stack die neuen Daten verwirft. Rechtzeitig heißt hier, dass z. B. beim Senden von 11 Telegrammen mit je 148 Bytes das letzte Telegramm verloren geht, wenn keines der anderen Telegramme von der Applikation abgeholt worden ist.

Bei der TCP-Kommunikation muss die Applikation dafür sorgen, dass die Nutzdaten zeitnah abgeholt werden. Ist das nicht der Fall können keine weiteren Daten mehr empfangen werden. An der Stelle werden keine Nutzdaten verworfen, da TCP eine Flusssteuerung beinhaltet. Der Kommunikationspartner sendet in dem Zustand keine weitern Daten mehr und signalisiert dies seiner Anwendung.

UDP- und TCP-Kommunikation

Eine Kommunikationsverbindung ist charakterisiert durch zwei Endpunkte. Ein Endpunkt ist ein geordnetes Paar aus IP Adresse und Port. Dabei stellt in allgemeinen ein Endpunkt einen Server und der andere Endpunkt einen Client dar. Der Server stellt über einen Port einen Dienst zur Verfügung. Der Client nutzt den zur Verfügung gestellten Dienst des Servers.

Maximale Anzahl der möglichen TCP Verbindungen

Beispielhaft finden Sie in der folgenden Tabelle die Anzahl der möglichen Kommunikationsverbindungen einer SIMOTION-CPU als Client. Die Werte beziehen sich auf ein lokales Netzwerk ohne weitere Fremdlast und eine SIMOTION D435 als Server. Abhängig von vorhandener Projektierung, Hardware und Netzauslastung/-topologie kann die Anzahl der möglichen Kommunikationsverbindungen variieren.

Tabelle 6- 4 Kommunikationsverbindungen in Abhängigkeit der SIMOTION-CPU

SIMOTION-CPU (Client)	Anzahl Kommunikationsverbindungen
C240	45
D410	45
D435	75
P350	40

Hinweis

Um eine sichere zyklische Kommunikation zu gewährleisten, sollten Sie auf Standardmechanismen wie z. B. unter PROFIBUS DP oder PROFINET IRT zurückgreifen, um mögliche kritische Ausfälle zu vermeiden, die bei UDP- bzw. TCP-Kommunikation auftreten können.

6.4 Vorbereitungen für die Verbindungs-Projektierung zwischen SIMOTION und SIMATIC S7

Voraussetzungen

Bevor eine TCP-/IP- bzw. eine UDP-Kommunikations-Verbindung zwischen einer SIMATIC S7-Station mit einem Ethernet-CP und einem SIMOTION Gerät angelegt werden kann, wird vorausgesetzt, dass die SIMATIC S7-Station und das SIMOTION Gerät im gleichen Projekt angelegt sind. (Multiprojekte werden in der Version V4.1 von SIMOTION nicht unterstützt.)

Weitere Voraussetzung ist es, dass in der SIMATIC-Station der Ethernet-CP konfiguriert und ein Ethernet als Netz im Projekt vorhanden sein muss. Außerdem müssen die beiden Kommunikations-Teilnehmer mit dem Netz verbunden und die Adress-Vergabe erfolgt sein.

Die Kommunikations-Verbindung wird in NetPro projektiert. Der Zugang zu NetPro ist über mehrere Wege möglich:

- Im SIMATIC Manager, dem SIMOTION SCOUT und HW Konfig kann über einen Menübutton NetPro gestartet werden.
- Im SIMOTION SCOUT über das Menü Projekt -> NetPro öffnen oder im SIMATIC Manager über das Menü Extra -> Netz konfigurieren kann ebenfalls NetPro geöffnet werden.
- Eine weitere Möglichkeit ist es NetPro über das Objekt Verbindungen innerhalb einer S7 CPU zu öffnen (siehe auch nachfolgende Abbildung). Vorteil von dieser Variante ist es, dass sofort die Verbindungstabelle der entsprechenden SIMATIC S7-Station geöffnet wird.

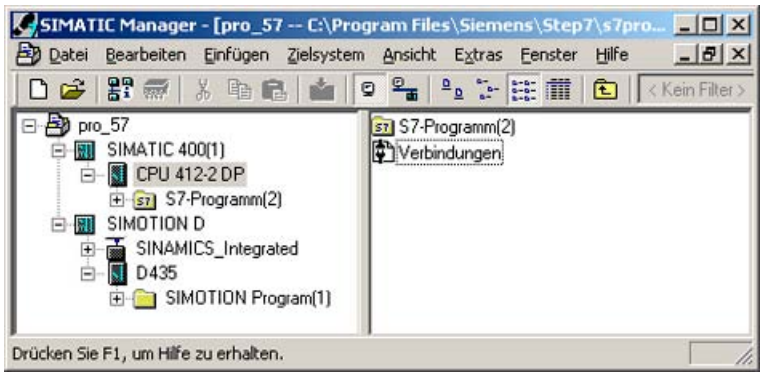


Bild 6-3 Darstellung des Objektes "Verbindungen" im SIMATIC Manager

6.5 Kommunikations-Verbindung zwischen SIMATIC mit Ethernet-CP und einem SIMOTION Gerät projektieren

6.5.1 Kommunikations-Verbindung zwischen SIMATIC mit Ethernet-CP und einem SIMOTION Gerät projektieren

Vorgehensweise

Um die Kommunikations-Verbindung anzulegen und zu projektieren, muss in NetPro die Verbindungstabelle der S7-Station dargestellt sein. Hierzu wird die S7 CPU innerhalb der S7-Station markiert. Im unteren Arbeitsbereich von NetPro wird nun die Verbindungstabelle angezeigt. Für das SIMOTION Gerät kann in NetPro keine Verbindungstabelle angezeigt werden.

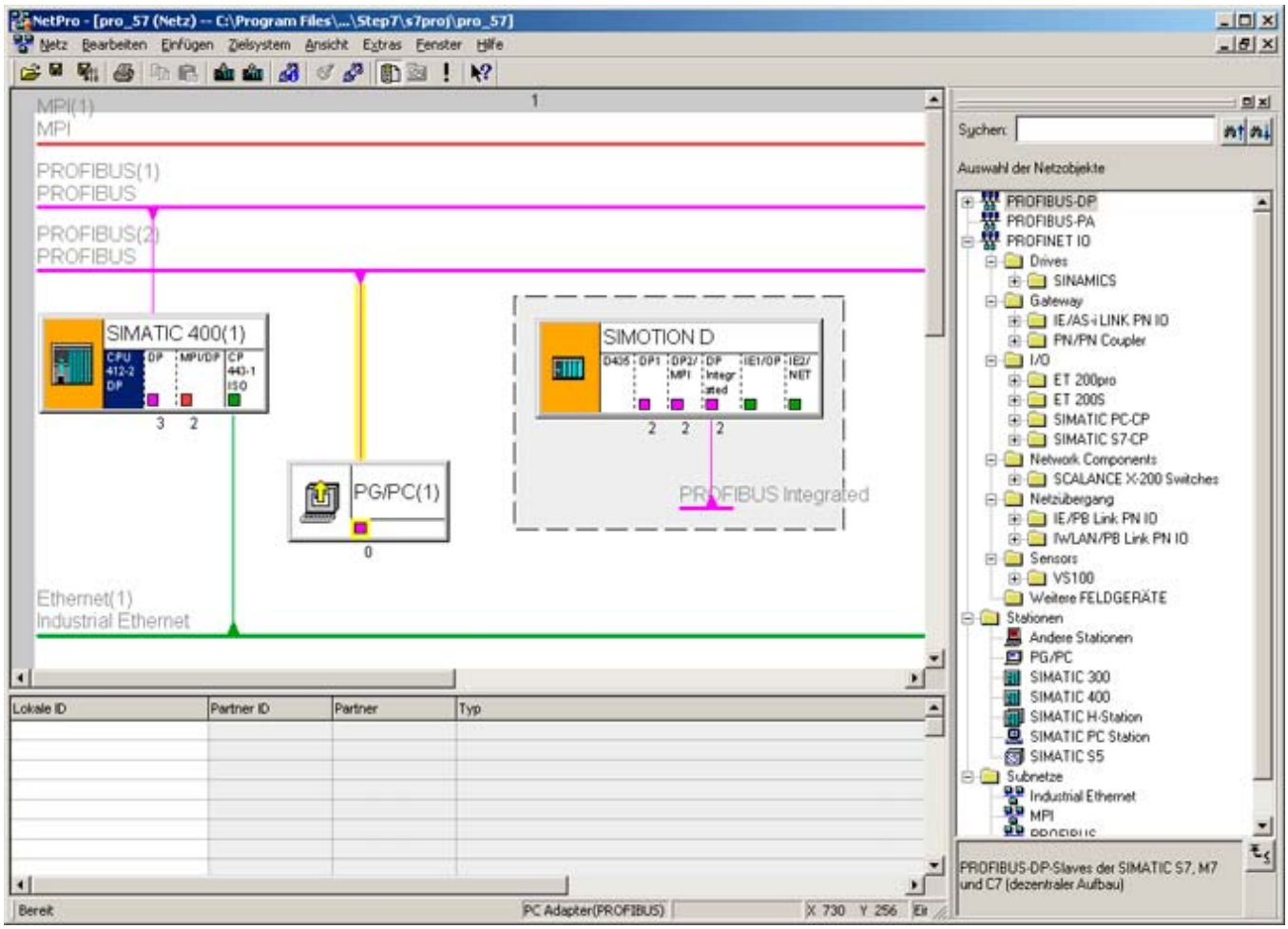


Bild 6-4 Markierte S7 CPU und die dazugehörige Verbindungstabelle

Durch Doppelklick auf eine leere Zeile in der Verbindungstabelle wird der Dialog **Neue Verbindung einfügen** zum Einfügen einer neuen Kommunikations-Verbindung geöffnet. Der Dialog ist für TCP-/IP- und UDP-Verbindung gleich. Er kann bei markierter S7 CPU auch über das Kontextmenü, über das Menü "Einfügen" -> "Neue Verbindung…" oder durch Klicken auf den Button in der Menüleiste erreicht werden.

6.5.2 TCP-/IP-Verbindung

Vorgehensweise

Im Dialog **Neue Verbindung einfügen** wird im Feld **Verbindungspartner** für die Kommunikation zwischen einem S7 CP und einem SIMOTION Gerät die Einstellung "(unspezifiziert)" beibehalten, da der Kommunikations-Partner – das SIMOTION Gerät – in der Auswahl nicht angeboten wird. Im Feld **Verbindung** wird der gewünschte Verbindungs-Typ **TCP-Verbindung** ausgewählt.

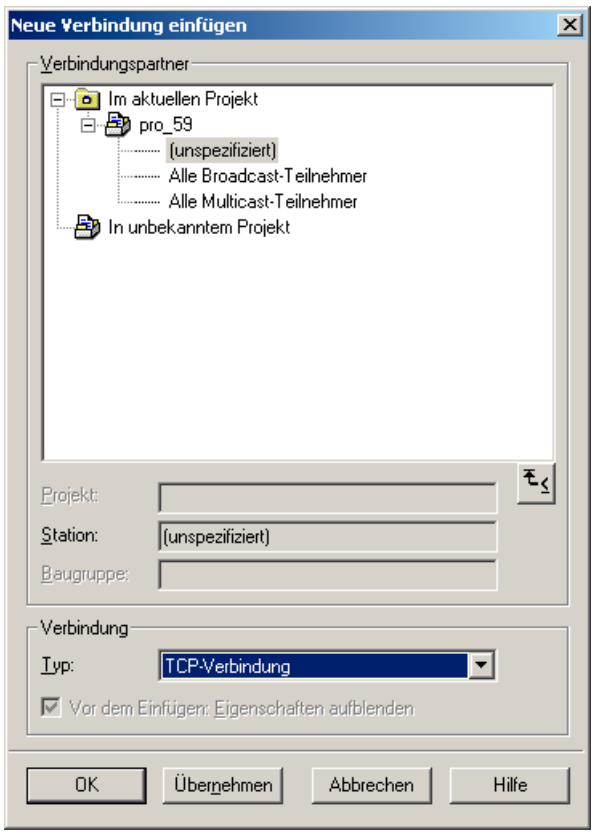


Bild 6-5 Dialog **Neue Verbindung einfügen** mit ausgewählter TCP-/IP-Verbindung

Wird der Dialog **Neue Verbindung einfügen** mit **OK** oder **Übernehmen** verlassen erscheint zunächst eine Hinweismeldung, die darauf verweist, dass auch Verbindungen über Subnetze hinweg möglich sind und eventuell die Routeradressen überprüft werden sollten. Nach dem Quittieren dieser Hinweismeldung öffnet sich für eine TCP-/IP-Verbindung der Dialog **Eigenschaften - TCP - Verbindung**.

Im Register **Adressen** sind die IP-Adresse und auch der Port für den lokalen Kommunikationspartner schon vorbelegt. Für den fernen Kommunikations-Partner sind die Angaben noch zu vervollständigen. In das Feld IP (DEZ) ist die IP-Adresse des SIMOTION Gerätes einzutragen.

Im Feld Port (DEZ) ist ein Port anzugeben, der auf dem SIMOTION Gerät für diese Kommunikations-Verbindung vom Anwender festgelegt wurde. Für den Port auf S7-Seite sind Randbedingungen einzuhalten, d.h. auf S7-Seite sollte ein Port zwischen 2000 und 5000 ausgewählt werden. Zweckmäßigerweise behält man den vorgeschlagenen Port bei.

Bild 6-6 Dialog Eigenschaften - TCP-Verbindung" - Register "Adressen"

Im Register **Allgemein** können im Feld **Bausteinparameter** die für die Parametrierung der SEND/RECEIVE-Schnittstelle wichtigen Bausteinparameter, über die die Verbindung im Anwenderprogramm referenziert werden kann, entnommen werden. Mit **ID** wird der Kommunikationsverbindung eine eindeutige Referenz zugewiesen. Zudem wird als "LADDR" die Adresse des CP angegeben.

Über die Anwahl der Check-Box **Aktiver Verbindungsaufbau** kann festgelegt werden, ob der Verbindungsaufbau von der S7-Station aus erfolgen soll. Wenn auf S7-Seite ein aktiver Verbindungsaufbau angewählt ist, so müssen auf SIMOTION-Seite für den Verbindungsauf- und -abbau die Aufrufe _tcpOpenServer und _tcpCloseServer im Anwender-Programm verwendet werden.

Wird auf S7-Seite jedoch kein aktiver Verbindungsaufbau angewählt, so müssen auf SIMOTION-Seite im Anwender-Programm für den Verbindungsauf- und -abbau die Aufrufe _tcpOpenClient und _tcpCloseConnection verwendet werden.

Ist eine Verbindung – egal von welchem Kommunikationspartner aus – erst einmal aufgebaut, so kann über sie von beiden Kommunikationspartner aus sowohl gesendet als auch empfangen werden.

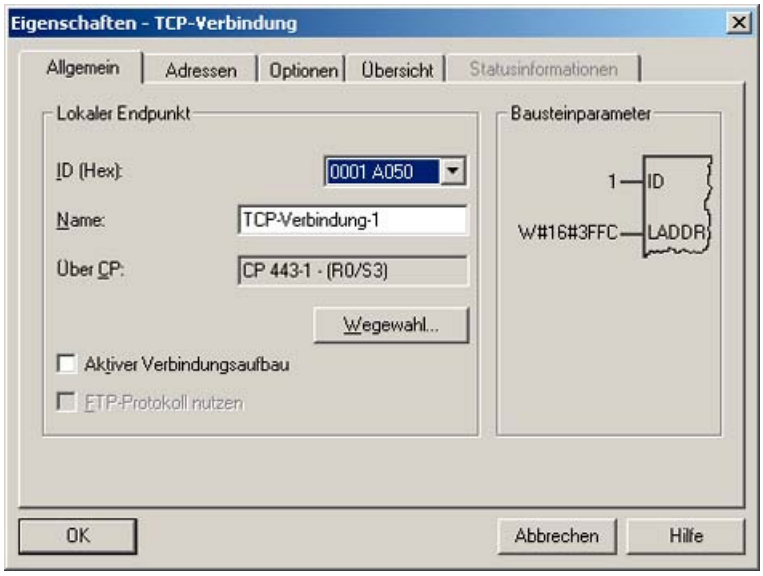


Bild 6-7 Dialog **Eigenschaften - TCP-Verbindung** - Register **Allgemein**

Wird der Dialog **Eigenschaften - TCP-Verbindung** nun mit **OK** verlassen, so muss noch der Dialog **Neue Verbindung einfügen** durch Klicken des Button **Schließen** geschlossen werden, um die Verbindungs-Projektierung endgültig abzuschließen. Möchte man weitere Verbindungen projektieren, so ist dies durch Anwahl des gewünschten Verbindungs-Typs und anschließendem Klicken auf den Button **Übernehmen** möglich.

Nach erfolgter Projektierung der Kommunikations-Verbindung stehen die Parameter für den Aufruf der Kommunikations-Funktionen im S7- und im SIMOTION-Anwenderprogramm fest. Für das S7-Anwenderprogramm werden die im Register **Allgemein** angegebenen Bausteinparameter benötigt. Für das SIMOTION-Anwenderprogramm werden die IP-Adresse des SIMATIC-CP, der Port auf S7-Seite (Lokaler Port im Register **Adresse**) und der Port auf SIMOTION-Seite (Partner-Port im Register **Adresse**) benötigt.

6.5.3 UDP-Verbindung

Vorgehensweise

Im Dialog **Neue Verbindung einfügen** wird im Feld **Verbindungspartner** für die Kommunikation über eine UDP-Verbindung zwischen dem SIMATIC S7 CP und einem SIMOTION Gerät die Einstellung "(unspezifiziert)" beibehalten, da der Kommunikations-Partner – das SIMOTION Gerät – in der Auswahl nicht angeboten wird. Im Feld **Verbindung** wird der gewünschte Verbindungs-Typ **UDP-Verbindung** ausgewählt.

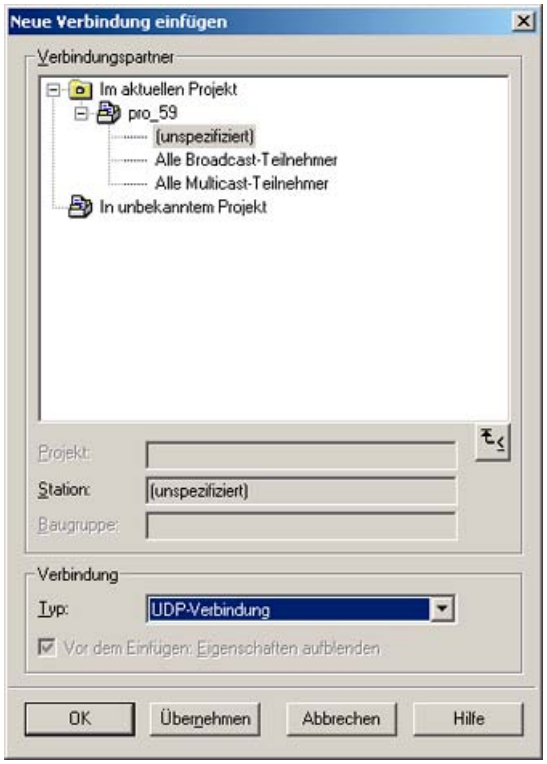


Bild 6-8 Dialog **Neue Verbindung einfügen** mit ausgewählter UDP-Verbindung

Wird der Dialog **Neue Verbindung einfügen** mit **OK** oder **Übernehmen** verlassen, erscheint ein Hinweis, dass Verbindungen über Subnetze hinweg möglich sind und eventuell die Routeradressen überprüft werden sollten. Nach dem Quittieren dieses Hinweises öffnet sich für eine UDP-Verbindung der Eigenschaften-Dialog (siehe **Dialog Eigenschaften - UDP-Verbindung - Register Adresse**).

Im Register **Adressen** sind die IP-Adresse und der Port für den lokalen Kommunikationspartner schon vorbelegt. Für den fernen Kommunikations-Partner sind die Angaben noch zu vervollständigen. In das Feld **IP (DEZ)** ist die IP-Adresse des SIMOTION Gerätes einzutragen. Im Feld **Port (DEZ)** ist ein Port anzugeben, der auf dem SIMOTION Gerät für diese Kommunikations-Verbindung vom Anwender festgelegt wurde.

Für den Port auf S7-Seite sind Randbedingungen einzuhalten, d.h. auf S7-Seite sollte ein Port größer 2000 ausgewählt werden. Zweckmäßigerweise behält man den vorgeschlagenen Port bei. Das Feld **Adressvergabe am Baustein** wird nicht ausgefüllt.

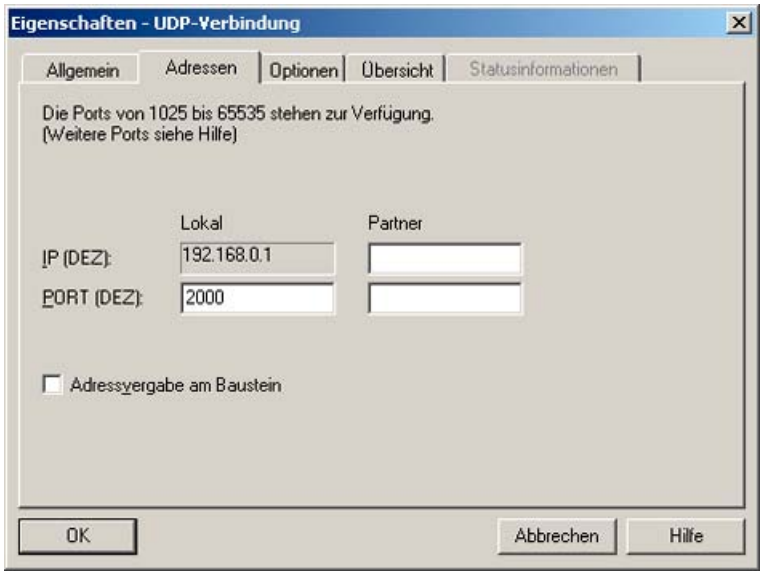


Bild 6-9 Dialog **Eigenschaften - UDP-Verbindung** - Register **Adresse**

Im Register **Allgemein** können im Feld **Bausteinparameter** die für die Parametrierung der SEND/RECEIVE-Schnittstelle wichtigen Bausteinparameter, über die die Verbindung im Anwenderprogramm referenziert werden kann, entnommen werden. Mit **ID** wird der Kommunikationsverbindung eine eindeutige Referenz zugewiesen. Zudem wird als "LADDR" die Adresse des CP angegeben.

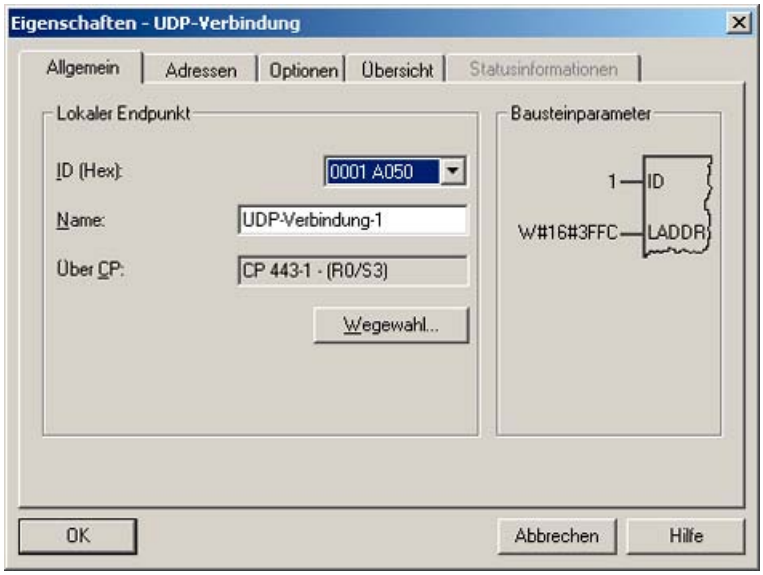


Bild 6-10 Dialog **Eigenschaften - UDP-Verbindung** - Register **Allgemein**

Wird der Dialog **Eigenschaften - UDP-Verbindung** nun mit **OK** verlassen, so muss noch der Dialog **Neue Verbindung einfügen** durch Klicken des Button **Schließen** geschlossen werden, um die Verbindungs-Projektierung endgültig abzuschließen. Möchte man weitere Verbindungen projektieren, so ist dies mit Anwahl des gewünschten Verbindungs-Typs und anschließendem Klicken auf den Button **Übernehmen** möglich.

Nach erfolgter Projektierung der Kommunikations-Verbindung stehen die Parameter für den Aufruf der Kommunikations-Funktionen im S7- und im SIMOTION-Anwenderprogramm fest. Für das S7-Anwenderprogramm werden, wie schon erwähnt, die im Register **Allgemein** angegebenen Bausteinparameter benötigt. Für das SIMOTION-Anwenderprogramm werden die IP-Adresse des SIMATIC-CP, der Port auf S7-Seite (Lokaler Port im Register **Adresse**) und der Port auf SIMOTION-Seite (Partner-Port im Register **Adresse**) benötigt.

6.6 Kommunikations-Verbindung zwischen einer SIMATIC-CPU mit integrierter Ethernet-Schnittstelle und einem SIMOTION Gerät anlegen

Das Anlegen einer Kommunikations-Verbindung zwischen einer SIMATIC-CPU mit integrierter Ethernet-Schnittstelle und einem SIMOTION Gerät läuft prinzipbedingt anders ab als die Projektierung einer Kommunikations-Verbindung zwischen einer SIMATIC-CPU mit Ethernet-CP und einem SIMOTION Gerät.

Über NetPro kann im Falle der Kommunikation zwischen einem SIMOTION Gerät und einer SIMATIC-CPU mit integrierter Ethernet-Schnittstelle – analog zur Kommunikation zwischen zwei SIMOTION Geräten – keine Verbindung eingefügt werden.

Lediglich in HW Konfig werden der SIMATIC CPU und dem SIMOTION Gerät IP-Adressen zugewiesen.

Die weiteren notwendigen Parameter, um die Kommunikations-Verbindung aufzubauen, werden vom Anwender für die beiden beteiligten Kommunikations-Partner festgelegt und beim Baustein-Aufruf auf S7-Seite bzw. beim Aufruf der Kommunikations-Funktion auf SIMOTION-Seite übergeben.

Auf S7-Seite werden die für den Aufbau der Kommunikations-Verbindung benötigten Parameter mittels eines Datenbausteins, der einen festgelegten Aufbau hat, übergeben. Der Aufbau des Datenbausteins wird unter SIMATIC S7 Funktionsbausteine und SIMOTION-Funktionen zum Aufbau einer TCP/IP-Verbindung (Seite 124) erläutert.

Eine Kommunikation zwischen einer SIMATIC-CPU mit integrierter Ethernet-Schnittstelle und einem SIMOTION Gerät über UDP ist nicht möglich!

6.7 Anwendung der Funktionen und Funktionsbausteine im Anwenderprogramm

6.7.1 Projektierungs-Flußdiagramm und Allgemeines

Vorgehensweise

In den folgenden Abbildungen ist dargestellt, wie bei der Projektierung einer Kommunikationsverbindung zwischen einer SIMATIC-Station und einer SIMOTION Station vorgegangen wird. Zusätzlich kann dem Flussdiagramm entnommen werden, in welchen Kapiteln die einzelnen beschriebenen Schritte detaillierter beschrieben sind.

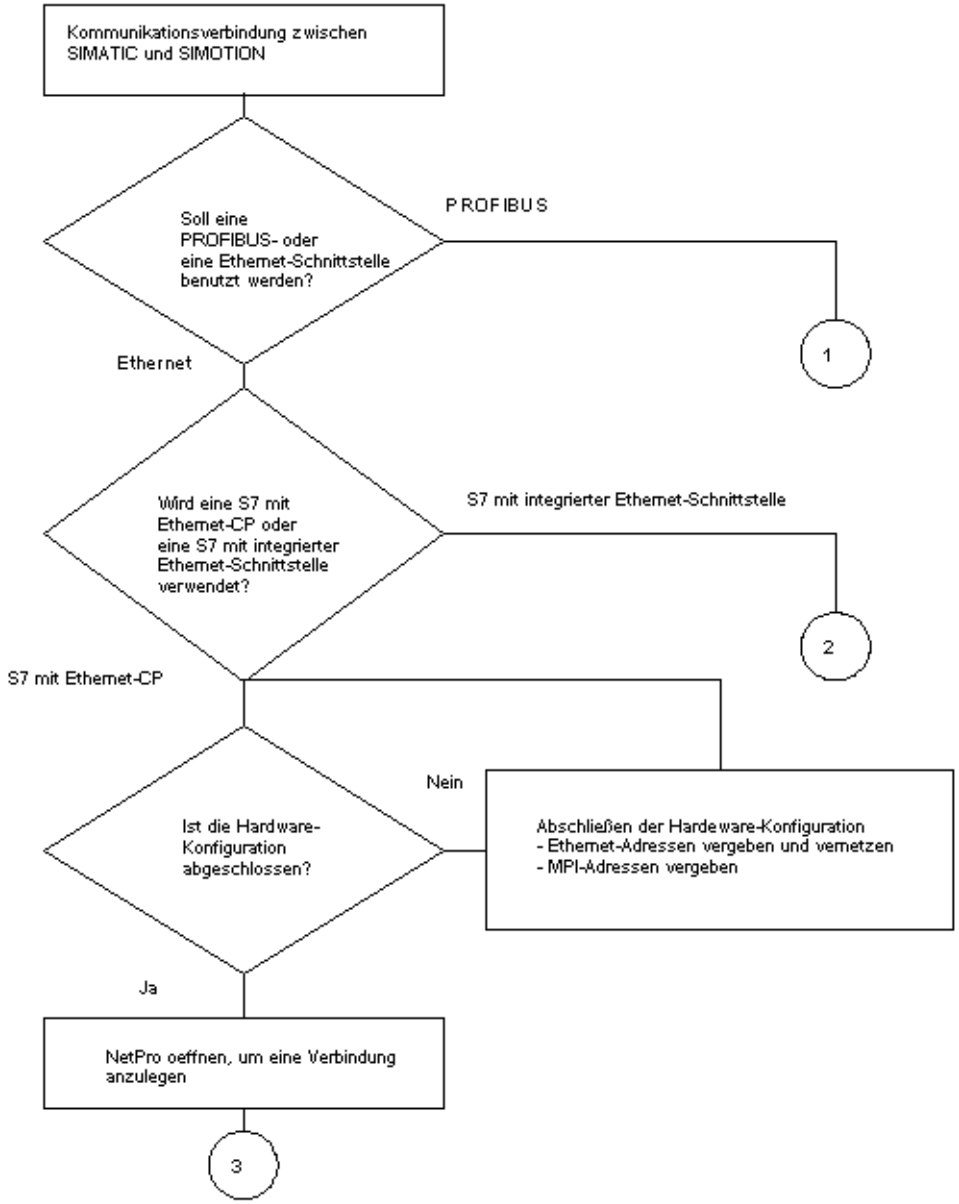


Bild 6-11 Flussdiagramm zur Projektierung einer Kommunikationsverbindung: Auswahl des Kommunikationsprotokolls und der SIMATIC-Station

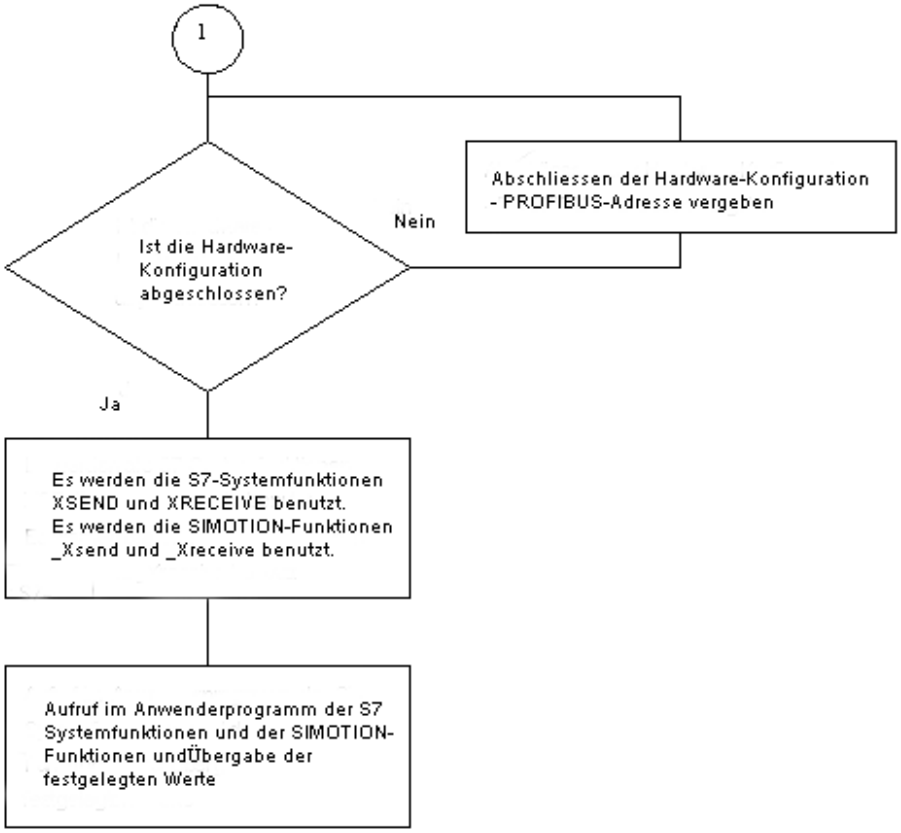


Bild 6-12 Flussdiagramm zur Projektierung einer Kommunikationsverbindung - Fortsetzung: PROFIBUS-Verbindung

Hinweis

Die Beschreibungen zu den S7-Funktionen sowie den SIMOTION Funktionen finden Sie in S7-Systemfunktionen und SIMOTION-Funktionen für eine PROFIBUS-Verbindung (Seite 90).

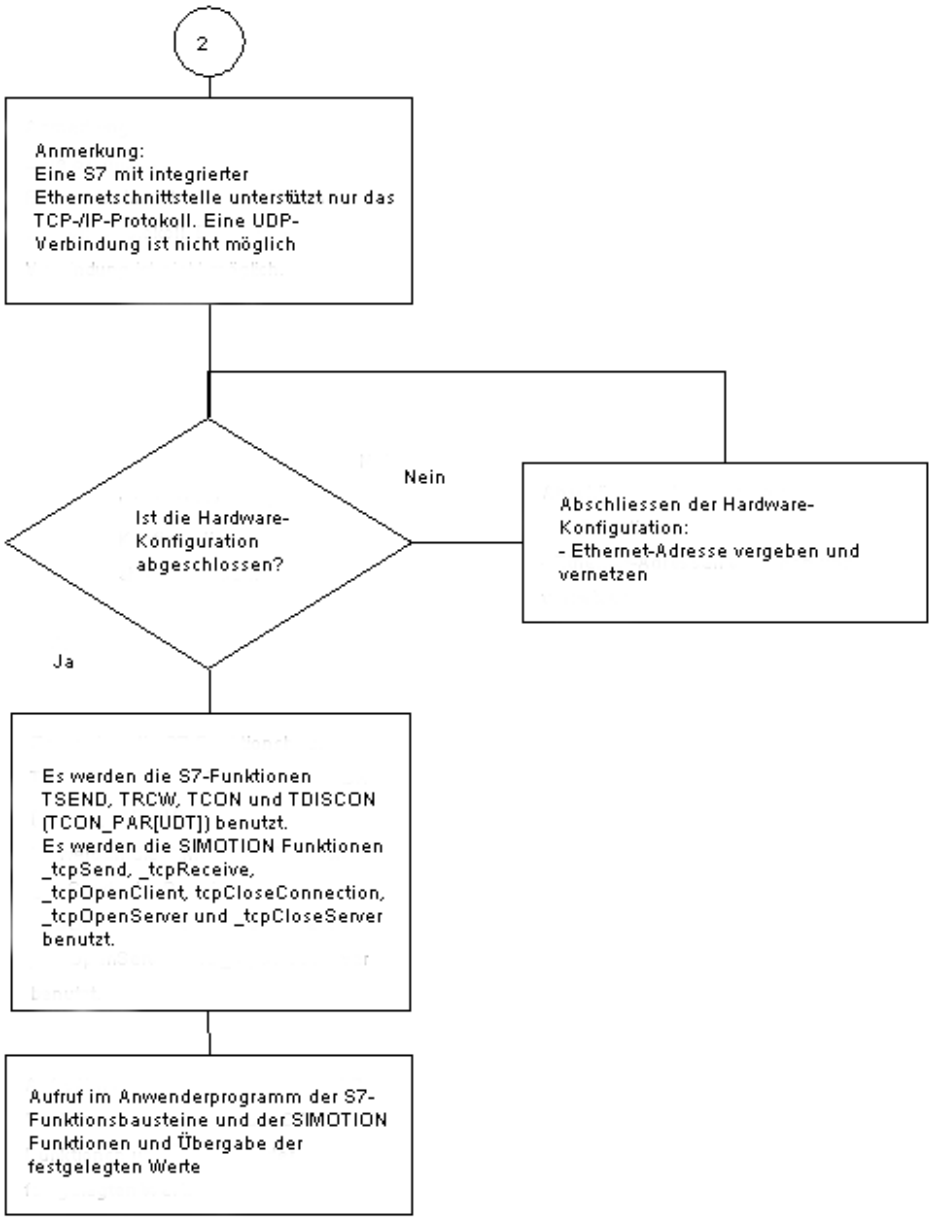


Bild 6-13 Flussdiagramm zur Projektierung einer Kommunikationsverbindung - Fortsetzung: TCP-/IP-Verbindung (S7 mit integrierter Ethernetschnittstelle)

Hinweis

Die Beschreibungen zu den S7-Funktionen sowie den SIMOTION Funktionen finden Sie in S7 Funktionsbausteine und SIMOTION-Funktionen für eine TCP-/IP-Verbindung beim Einsatz einer S7-Station mit integrierter Ethernetschnittstelle (Seite 111).

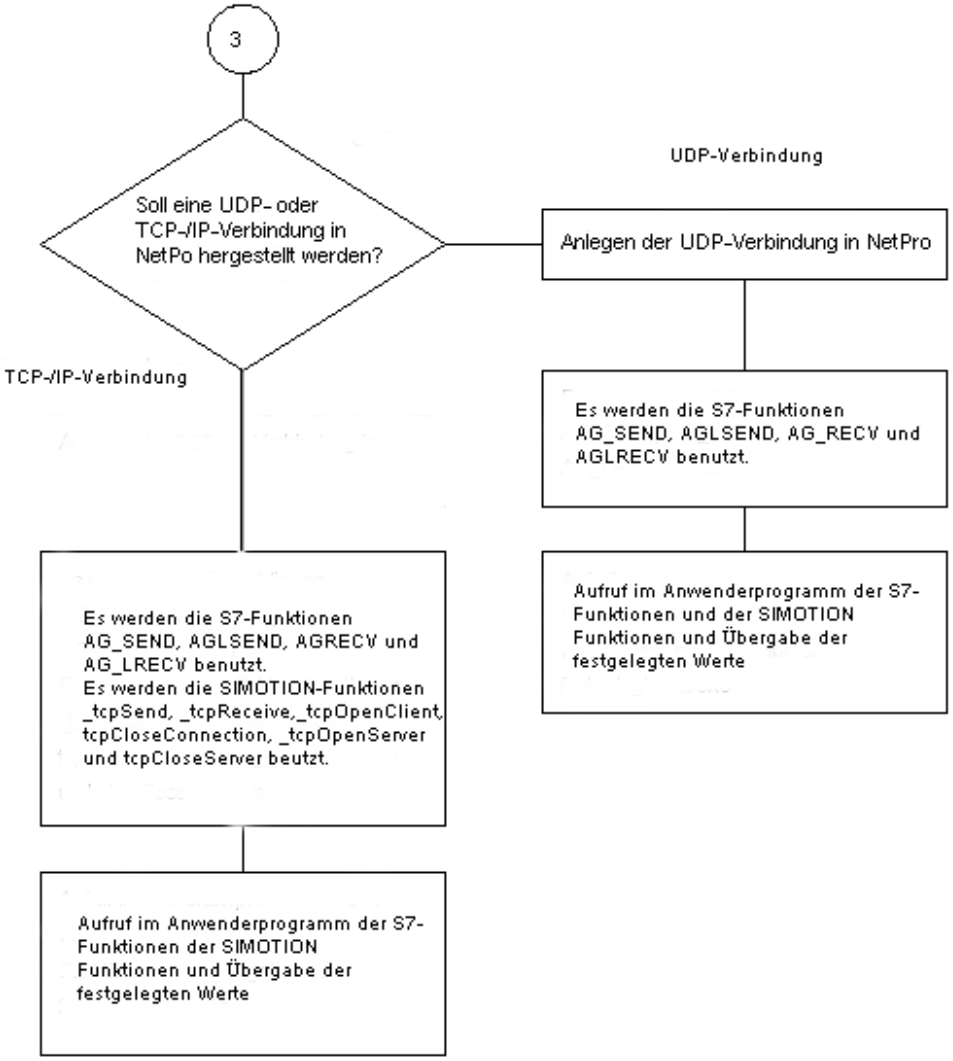


Bild 6-14 Flussdiagramm zur Projektierung einer Kommunikationsverbindung - Fortsetzung: TCP-/IP- und UDP-Verbindung (S7 mit Ethernet-CP)

Hinweis

Die Beschreibungen zu den S7-Funktionen sowie den SIMOTION Funktionen finden Sie in *S7- und SIMOTION-Funktionen für eine TCP-/IP-Verbindung beim Einsatz einer S7-Station mit Ethernet-CP*.

Die vorgegebenen bzw. ermittelten Parameter müssen Sie anschließend bei der Parametrierung der Schnittstelle im Anwenderprogramm zu verwenden, wenn sie folgendes durchgeführt haben:

- Sie haben eine Kommunikationsverbindung zwischen einer SIMATIC-CPU mit Ethernet-CP und einem SIMOTION-Gerät projektiert (siehe Kommunikations-Verbindung zwischen SIMATIC mit Ethernet-CP und einem SIMOTION Gerät projektieren (Seite 105))
- Sie haben die Parameter für den Aufbau einer Kommunikationsverbindung per Bausteinaufruf festgelegt (siehe Kommunikations-Verbindung zwischen einer SIMATIC-CPU mit integrierter Ethernet-Schnittstelle und einem SIMOTION Gerät anlegen (Seite 111))

Auch die für Kommunikation über PROFIBUS festgelegten Parameter sind für die Parametrierung der Anwender-Schnittstelle zu verwenden.

6.7.2 S7- und SIMOTION-Funktionen für eine TCP-/IP-Verbindung beim Einsatz einer S7-Station mit Ethernet-CP

6.7.2.1 Einleitung

Im Folgenden soll die Parametrierung der SIMATIC S7- bzw. der SIMOTION-Funktionen für eine TCP-/IP-Verbindung, die für den Einsatz mit einer SIMATIC S7-Station mit Ethernet-CP angelegt wurde, erläutert werden.

6.7.2.2 S7-Funktionen

Je nach Baureihe der S7-Station (S7-300 oder S7-400) stehen für den beschriebenen Anwendungsfall in Sende- und Empfangsrichtung je zwei Funktionen zur Verfügung.

In Senderichtung sind dies die Funktionen FC5 (AG_SEND) und FC50 (AG_LSEND). Im Folgenden ist dargestellt wie ein Aufruf im Anwenderprogramm und die jeweilige Parametrierung aussehen.

Tabelle 6- 5 Programmbeispiel

```
CALL "AG_Send"
 Act :=M0.0
 ID :=1
 LADDR :=W#16#3FFD
 SEND :=P#DB100.DBX0.0 BYTE 1000
 LEN :=1000
 DONE :=M0.1
 ERROR :=M0.2
 STATUS :=MW10
CALL "AG_LSEND"
 ACT :=M0.0
 ID :=1
 LADDR :=W#16#3FFD
 SEND :=P#DB100.DBX0.0 BYTE 1000
 LEN :=1000
```

```
 DONE :=M0.1
 ERROR :=M0.2
 STATUS :=MW1.0
```

Der Aufbau und somit auch die Parametrierung sind identisch. Es sollen daher die wichtigen Parameter für beide Funktionen zusammen erläutert werden.

Die Parameter ID und LADDR werden beim Anlegen der Verbindung in NetPro angezeigt und müssen für diese Verbindung beim Aufruf der Funktionen übergeben werden. Über ACT wird das Senden angestoßen. Der Parameter SEND definiert die Sendedaten und über LEN wird die zu sendende Länge festgelegt. Die Parameter DONE, ERROR und STATUS dienen der Diagnose bzw. liefern den Status zurück, den der Sendeauftrag hat.

Auch in Empfangsrichtung existieren zwei Funktionen FC6 (AG_RECV) und FC60 (AG_LRECV). Im Folgenden ist dargestellt wie ein Aufruf im Anwenderprogramm und die jeweilige Parametrierung aussehen.

Tabelle 6- 6 Programmbeispiel

```
Call "AG_RECV"
 ID :=1
 LADDR :=W#16#3FFD
 RECV :=P#DB110.DBX0.0 BYTE 1000
 NDR :=M2.0
 ERROR :=M2.1
 STATUS :=MW14
 LEN :=MW16
Call "AG_LRECV"
 ID :=1
 LADDR :=W#16#3FFD
 RECV :=P#DB110.DBX0.0 BYTE 1000
 NDR :=M2.0
 ERROR :=M2.1
 STATUS :=MW14
 LEN :=MW16
```

Auch in Empfangsrichtung sieht man, dass der Aufbau und somit auch die Parametrierung der Funktionen exakt identisch sind. Daher sollen die wichtigen Parameter für beide Funktionen zusammen erläutert werden.

Beim Anlegen der Verbindung in NetPro werden die Parameter ID und LADDR von NetPro festgelegt und müssen für diese Verbindung beim Aufruf der Funktionen übergeben werden. Der Parameter RECV gibt den Datenbereich an, in dem die empfangenen Daten abgelegt werden. Über NDR wird dem Anwender mitgeteilt, wenn neue Daten empfangen wurden (1: neue Daten). Der Parameter LEN gibt an in welcher Länge in Byte neue Daten empfangen wurden. Über die Parameter ERROR und STATUS wird eine Diagnose bzw. der Status zurück geliefert, den der Empfangsaufruf hat.

6.7.2.3 SIMOTION-Funktionen

Für den Anwendungsfall einer TCP-/IP-Verbindung zwischen einer S7-Station mit Ethernet-CP und einem SIMOTION Gerät stehen auf SIMOTION-Seite insgesamt sechs Funktionen zur Verfügung, die allerdings nicht alle gleichzeitig benötigt werden.

Je nach Projektierung des Verbindungsaufbaus auf S7-Seite werden unterschiedliche Funktionen auf SIMOTION-Seite verwendet, um die Verbindung auf- und abzubauen.

Wurde auf S7-Seite ein aktiver Verbindungsaufbau festgelegt, so wird auf SIMOTION-Seite die Verbindung mit der Funktion _tcpopenserver aufgebaut und mit der Funktion _tcpcloseserver wieder abgebaut.

Wurde auf S7-Seite kein aktiver Verbindungsaufbau projektiert, so werden zum Verbindungsaufbau auf SIMOTION-Seite die Funktion _tcpopenclient und zum Verbindungsabbau die Funktion _tcpcloseconnection verwendet.

Das Senden und Empfangen läuft unabhängig von dem projektierten Verbindungsauf- und -abbau über die Funktionen _tcpsend und _tcprecieve. Es ist allerdings erforderlich, vor dem Senden bzw. Empfangen eine Verbindung aufzubauen.

```
RetVal_TCPOpenClient  :=_TCPOpenClient
    (port             :=D435_Port,
    serveraddress     :=S7_IP_Adresse,
    serverport        :=S7_Port,
    nextcommand       :=WHEN_COMMAND_DONE);
```

Aufruf-Beispiel der SIMOTION-Funktion_tcpopenclient

Soll eine Verbindung zu einer S7-Station aufgebaut werden, für die auf S7-Seite kein aktiver Verbindungsaufbau angewählt wurde, so wird im SIMOTION-Anwenderprogramm die Funktion _tcpopenclient aufgerufen. Beim Aufruf wird der Funktion für den Parameter port der lokal vergebene SIMOTION-Port übergeben. Der Parameter serveraddress ist die IP-Adresse der S7-Station, die in einem Array übergeben wird. Im Paramter serverport wird der Funktion die als lokale Portnummer bezeichnete Portnummer übergeben. Mit dem Parameter nextcommand wird das Verhalten der Funktion in Bezug auf das Weiterschalten beim Aufruf parametriert. Es gibt zwei Einstellmöglichkeiten: IMMEDIATELY und WHEN_COMMAND_DONE. Beim ersten Wert wird unmittelbar und beim zweiten Wert erst nach Beendigung des Befehls weitergeschaltet.

An das Anwenderprogramm wird bei Aufruf der Funktion _tcpopenclient eine Struktur zurückgegeben, die folgende Parameter enthält. Über den Parameter functionResult kann der Status des Verbindungsaufbaus abgefragt werden. Der Parameter connectionid dient als (Eingangs-)Parameter für den Aufruf der Funktionen _tcpsend, _tcprecieve und _tcpcloseconnection und ordnet den vorgenannten Funktionen eindeutig eine TCP-/IP-Verbindung zu. In den folgenden Aufruf-Beispielen wird auf diesen Rückgabewert Bezug genommen.

```
RetVal_TCPSend      := _TCPSend
    (connectionid   :=RetVal_TCPOpenClient.ConnctionID,
    nextcommand     :=WHEN_COMMAND_DONE,
    datalength      :=Soll_Sende_Datenlaenge,
    data            :=TCP_Sende_Daten);
```

Aufruf-Beispiel der SIMOTION-Funktion_tcpsend

Für das Senden von Daten von SIMOTION zur SIMATIC wird im SIMOTION-Anwenderprogramm die Funktion _tcpsend aufgerufen.

Die Parameter, die beim Aufruf der Funktion übergeben werden müssen, sollen im Folgenden erläutert werden. Für den Parameter connectionid wird der Rückgabewert connectionid der Funktionen _tcpopenclient oder _tcpopenserver übergeben - je nach dem von welcher Station aus die aktive Verbindungsaufforderung gestartet wurde, um eindeutig festzulegen über welche Verbindung das Senden ausgeführt werden soll. Für diese Funktion wird ebenfalls mit dem Parameter nextcommand das Verhalten der Funktion in Bezug auf das Weiterschalten beim Aufruf parametriert. Es gibt zwei Einstellmöglichkeiten: IMMEDIATELY und WHEN_COMMAND_DONE. Beim ersten Wert wird unmittelbar oder beim zweiten Wert erst nach Beendigung des Befehls weitergeschaltet. Im Parameter datalength wird der Funktion mitgeteilt, wie viele Nutzdaten in Bytes übertragen werden sollen. Mit dem Parameter data wird angegeben, in welchem Nutzdatenbereich die Sendedaten liegen, die mit der Funktion übertragen werden sollen.

Der Rückgabewert der Funktion an das Anwenderprogramm hat den Datentyp DINT und über unterschiedliche Rückgabewerte wird auf Probleme bei der Ausführung der Funktion hingewiesen. Ebenso wird zurückgemeldet, wenn die Daten erfolgreich gesendet wurden.

```
RetVal_TCPReceive    := _TCPReceive
    (connectionid    :=RetVal_TCPOpenclient.ConnectionID,
     nextcommand     :=IMMEDIATELY,
     receivevariable :=TCP_Empfangs_Daten);
```

Aufruf-Beispiel der SIMOTION-Funktion_tcpreceive

Um Daten von einer SIMATIC-Station auf SIMOTION-Seite zu empfangen, wird im SIMOTION-Anwenderprogramm die Funktion _tcpreceive aufgerufen.

Beim Aufruf der Funktion werden ihr verschiedene Parameter übergeben. Für den Parameter connectionid wird der Rückgabewert connectionid der Funktionen _tcpopenclient oder _tcpopenserver übergeben - je nach dem von welcher Station aus die aktive Verbindungsaufforderung gestartet wurde, um eindeutig festzulegen über welche Verbindung das Empfangen ausgeführt werden soll. Für _tcpreceive wird ebenfalls mit dem Parameter nextcommand das Verhalten der Funktion in Bezug auf das Weiterschalten beim Aufruf parametriert. Es gibt zwei Einstellmöglichkeiten: IMMEDIATELY und WHEN_COMMAND_DONE. Beim ersten Wert wird unmittelbar bzw. beim zweiten Wert erst nach Beendigung des Befehls weitergeschaltet. Typischerweise werden bei der Übertragung verschiedener Nutzdatenlängen zwischen SIMATIC und SIMOTION ab einer Nutzdatenlänge von über 240 Bytes die Nutzdaten in Paketen nicht vorhersagbarer Größe übertragen.

Ist dies der Fall, so muss auf SIMOTION-Seite dafür Sorge getragen werden, dass die Nutzdaten vor der Auswertung und weiteren Verarbeitung in der korrekten Reihenfolge abgespeichert werden. Hierzu sollte der Parameter nextcommand auf IMMEDIATELY gesetzt werden. Mit dem Parameter receivevariable wird der Funktion mitgeteilt, in welchem Nutzdatenbereich die Empfangsdaten abgelegt werden sollen, die von der SIMATIC-Seite empfangen werden.

An das Anwenderprogramm wird bei Aufruf der Funktion _tcpreceive eine Struktur zurückgegeben, die folgende Parameter enthält. Über den Parameter functionresult kann der Status des Empfangens abgefragt werden. Parameter datalength meldet die Anzahl der empfangenen Nutzdatenbytes nach erfolgreichem Aufruf der Funktion _tcpreceive zurück.

```
RetVal_TCPCloseConnection    := _TCPCloseConnection
    (connectionid            :=RetVal_TCPOpenclient.ConnectionID
    );
```

Aufruf-Beispiel der SIMOTION-Funktion_tcpcloseconnection

Um eine Verbindung zu schließen, für die in der Verbindungskonfiguration kein aktiver Verbindungsaufbau auf S7-Seite angewählt wurde (d.h. es wurde kein Haken im Kästchen Aktiver Verbindungsaufbau gesetzt.), wird auf SIMOTION-Seite die Funktion _tcpcloseconnection aufgerufen.

Als einziger Übergabewert an die Funktion wird im Parameter connectionid der Rückgabewert connectionID der Funktion _tcpopenclient übergeben, um eindeutig festzulegen welche Verbindung abgebaut werden soll.

Der Rückgabewert der Funktion an das Anwenderprogramm hat den Datentyp DINT und weist auf Probleme bei der Ausführung der Funktion hin bzw. meldet zurück, wenn die Verbindung erfolgreich abgebaut wurde.

```
RetVal_TCPOpenServer                    := _TCPOpenServer
    &#i921; port:=D435_Port, backlog    := 5,
    nextcommand                         := WHEN_COMMAND_DONE);
```

Aufruf-Beispiel der SIMOTION-Funktion-tcpopenserver

Wird in der Verbindungskonfiguration ein aktiver Verbindungsaufbau auf S7-Seite angegeben (d.h. der Haken im Kästchen Aktiver Verbindungsaufbau wurde gesetzt.), dann ist die Verbindung auf SIMOTION-Seite für den Datenaustausch mit der Funktion _tcpopenserver zu öffnen.

Zur Parametrierung der Funktion wird für den Parameter port der lokal vergebene SIMOTION-Port übergeben. Als weiteren Parameter wird für backlog angegeben, wie viele parallele Verbindungsanforderungen maximal für diesen Port von anderen Steuerungen her zugelassen werden sollen. Auch für diese Funktion wird mit dem Parameter nextcommand das Verhalten der Funktion in Bezug auf das Weiterschalten beim Aufruf parametriert. Es gibt zwei Einstellmöglichkeiten: IMMEDIATELY und WHEN_COMMAND_DONE. Beim ersten Wert wird unmittelbar und beim zweiten Wert erst nach Beendigung des Befehls weitergeschaltet.

An das Anwenderprogramm wird bei Aufruf der Funktion _tcpopenserver eine Struktur zurückgegeben, die folgende Parameter enthält. Über den Parameter functionResult kann der Status des Verbindungsaufbaus abgefragt werden. Der Parameter connectionid dient als (Eingangs-)Parameter für den Aufruf der Funktionen _tcpsend und _tcprecieve und ordnet den vorgenannten Funktionen eindeutig eine TCP-/IP-Verbindung zu. In den oben genannten Aufruf-Beispielen wird auf diesen Rückgabewert Bezug genommen.

Die beiden folgenden Parameter, die in der Struktur an das Anwenderprogramm zurückgegeben werden, werden in NetPro vom Anwender projektiert und sind dem Anwender somit bekannt. Sie sollen trotzdem der Vollständigkeit halber angegeben werden. Der Parameter clientAddress liefert die IP-Adresse der S7-Station, von der aus die Verbindung aktiviert wird, in Form eines Arrays. Im Parameter clientPort wird die als lokale Portnummer bezeichnete Portnummer der S7-Station angegeben.

```
RetVal_TCPCloseServer:= _TCPCloseServer(port:= D435_Port);
```

Aufruf-Beispiel der SIMOTION-Funktion_tcpcloseserver

Soll eine Verbindung geschlossen werden, für die in der Verbindungskonfiguration ein aktiver Verbindungsaufbau auf S7-Seite angewählt wurde (d.h. der Haken im Kästchen Aktiver Verbindungsaufbau wurde gesetzt.), so ist die Verbindung auf SIMOTION-Seite mit der Funktion _tcpcloseserver zu beenden.

Als einziger Übergabewert an die Funktion wird für den Parameter port der lokal vergebene SIMOTION-Port übergeben.

Der Rückgabewert der Funktion an das Anwenderprogramm hat den Datentyp DINT und weist auf Fehler in der Parametrierung der Funktion hin bzw. meldet zurück, wenn der Port erfolgreich geschlossen wurde.

6.7.3 S7- und SIMOTION-Funktionen für eine UDP-Verbindung beim Einsatz einer S7-Station mit Ethernet-CP

6.7.3.1 Einleitung

Im Folgenden soll die Parametrierung der S7- bzw. der SIMOTION-Funktionen für eine UDP-Verbindung, die für den Einsatz mit einer S7-Station mit Ethernet-CP angelegt wurde, erläutert werden.

6.7.3.2 S7-Funktionen

Bei den für diesen Anwendungsfall benutzten Funktionen handelt es sich ebenfalls um die bereits beschriebenen S7-Funktionen FC5 (AG_SEND), FC50 (AG_LSEND), FC6 (AG_RECV) und FC60 (AG_LRECV).

Die Parameter ID und LADDR, die von NetPro für die Verbindung vergebenen werden, sind beispielhaft dargestellt. Alle weiteren Parameter zur Parametrierungen der vorgenannten Funktionen für eine UDP-Verbindung zwischen einer S7-Station mit Ethernet-CP und einem SIMOTION Gerät wurden bereits erläutert und können entsprechend nachgeschlagen werden.

6.7.3.3 SIMOTION-Funktionen

Für den Anwendungsfall einer UDP-Verbindung zwischen einer S7-Station mit Ethernet-CP und einem SIMOTION Gerät werden auf SIMOTION-Seite zwei Funktionen verwendet.

Das Versenden von Daten läuft über _udpSend. Sollen Daten auf SIMOTION-Seite empfangen werden, so wird die Funktion _udpReceive verwendet. In den folgenden Programmbeispielen ist der Aufruf und die Parametrierung dargestellt.

```
RetVal_UDPSend          :=_UDPSend(sourceport:= P350_Port,
    destinationaddress  :=S7 IP Adresse,
    destinationport     :=S7_Port,
```

```
    communicationmode   :=CLOSE_ON_EXIT,
    datalength          :=UDPDatalength_Send,
    data                :=UDPSendData);
```

Beispielhafter Aufruf der SIMOTION-Funktion_udpSend

Beim Aufruf der Funktion _udpSend wird für den Parameter sourceport der auf der SIMOTION-Seite vergebene Port übergeben. Der Parameter destinationaddress ist die IP-Adresse der S7-Station, die in einem Array übergeben wird. Die IP-Adresse der S7-Station kann in HW-Konfig konfiguriert und ausgelesen werden. Als destinationport wird der auf S7-Seite vergebene als "lokaler Port" bezeichnete Port übergeben. Über communicationmode kann der Anwender festlegen, ob die Kommunikations-Resourcen nach dem Senden frei gegeben (CLOSE_ON_EXIT) oder nicht freigegeben (DO_NOT_CLOSE_ON_EXIT) werden sollen. Die Parameter datalength und data geben die zu versendende Datenlänge bzw. den Bereich an, wo die Sendedaten abgelegt sind.

Über den Rückgabewert der Funktion kann man den Status des Sendeauftrags überprüfen.

```
RetVal_UDPReceive       :=_udpreceive(port:=P350_Port,
    communicationmode   :=CLOSE_ON_EXIT,
    nextcommand         :=WHEN_COMMAND_DONE,
    receivevariable     :=UDPReceiveData);
```

Aufruf-Beispiel der SIMOTION-Funktion_udpReceive

Wird die Funktion _udpReceive aufgerufen, so wird wieder für den Parameter port der als "Partner-Port" bezeichnete Port auf SIMOTION-Seite angegeben. Auch hier kann über communicationmode vom Anwender festgelegt werden, ob die Kommunikations-Resourcen nach dem Empfangen frei gegeben (CLOSE_ON_EXIT) oder nicht frei gegeben (DO_NOT_CLOSE_ON_EXIT) werden sollen.

Mit dem Parameter nextcommand wird das Verhalten der Funktion in Bezug auf das Weiterschalten beim Aufruf parametriert. Für diesen Parameter gibt es drei Einstellmöglichkeiten: IMMEDIATELY, WHEN_COMMAND_DONE und ABORT_CURRENT_COMMAND.

Bei den beiden ersten Werten wird entweder unmittelbar oder eben erst nach Beendigung des Befehls weitergeschaltet. Mit dem dritten Wert wird bei Übergabe derselben Port-Nummer wie beim vorhergehenden Aufruf der Funktion, die aktive Funktion abgebrochen. Der Parameter receivevariable gibt den Puffer an, in den die Empfangsdaten gelegt werden.

An das Anwenderprogramm wird bei Aufruf der Funktion _udpReceive eine Struktur zurückgegeben, die folgende Parameter enthält. Im Parameter functionResult kann der Status des Aufrufs der Empfangsfunktion abgefragt werden. Der Parameter sourceAddress ist ein Array, das die IP-Adresse der S7-Station enthält. Ebenso enthält der Parameter sourceport der Struktur den als lokalen Port bezeichneten Port der S7-Station. Im Parameter dataLength kann die Anzahl der empfangenen Nutzdatenbytes nach erfolgreichem Aufruf der Funktion _udpReceive ausgelesen werden.

6.7.4 S7-Funktionsbausteine und SIMOTION-Funktionen für eine TCP-/IP-Verbindung beim Einsatz einer S7-Station mit integrierter Ethernetschnittstelle

6.7.4.1 Einleitung

Im Folgenden soll die Parametrierung der S7-Funktionsbausteine bzw. der SIMOTION-Funktionen für eine TCP-/IP-Verbindung, die für den Einsatz mit einer S7-Station mit integrierter Ethernetschnittstelle angelegt wurde, näher erläutert werden.

6.7.4.2 S7-Funktionsbausteine

Für die Kommunikation zwischen einer SIMATIC-Station mit integrierter Ethernet-Schnittstelle und einem SIMOTION-Gerät stehen auf SIMATIC-Seite verschiedene Funktionsbausteine und ein UDT zur Verfügung. Für den Verbindungsaufbau wird der FB65 TCON aufgerufen. Zum Abbauen der Verbindung wird der Funktionsbaustein FB66 TDISCON verwendet. Das Senden und Empfangen der Nutzdaten wird mittels der Bausteine FB63 TSEND und FB64 TRCV durchgeführt.

Bei der Kommunikation zwischen einer SIMATIC-Station mit integrierter Ethernet-Schnittstelle und einem SIMOTION-Gerät entfällt die Verbindungsprojektierung in NetPro. Stattdessen wird die Verbindungsprojektierung im Anwenderprogramm vorgenommen. Auf SIMATIC-Seite wird dies durch Datenbausteine realisiert, die vom UDT65 TCON_PAR abgeleitet werden (siehe Bild unten). D.h. die verwendeten Datenbausteine müssen die Datenstruktur aus dem UDT65 enthalten.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	block_length	WORD	W#16#40	Länge 64 Byte
+2.0	id	WORD	W#16#0	xxx Verbindungs ID
+4.0	connection_type	BYTE	B#16#1	
+5.0	active_est	BOOL	FALSE	
+6.0	local_device_id	BYTE	B#16#2	
+7.0	local_tsap_id_len	BYTE	B#16#2	
+8.0	rem_subnet_id_len	BYTE	B#16#0	
+9.0	rem_staddr_len	BYTE	B#16#0	xxx 0: unspezifizierte Verb. 4: spezifizierte Verb.
+10.0	rem_tsap_id_len	BYTE	B#16#0	
+11.0	next_staddr_len	BYTE	B#16#0	
+12.0	local_tsap_id	ARRAY[1..16]	B#16#0	xxx lokaler Port
*1.0		BYTE		
+28.0	rem_subnet_id	ARRAY[1..6]	B#16#0	
*1.0		BYTE		
+34.0	rem_staddr	ARRAY[1..6]	B#16#0	xxx Remote IP-Adr.
*1.0		BYTE		
+40.0	rem_tsap_id	ARRAY[1..16]	B#16#0	
*1.0		BYTE		
+56.0	next_staddr	ARRAY[1..6]	B#16#9	
*1.0		BYTE		
+62.0	spare	WORD	W#16#0	
=64.0		END_STRUCT		

Bild 6-15 Deklarationssicht des UDT65 TCON_PAR im KOP-/AWL-/FUP-Editor

Der Aufbau und die Parametrierung des UDT65 soll im Folgenden näher erläutert werden. Bei der Parametrierung des UDT65 bzw. des sich daraus ableitenden Datenbausteins unterscheidet man prinzipiell zwei Fälle:

1. Der Verbindungsaufbau geht aktiv von der S7 Station aus.
2. Die S7 Station wartet passiv auf einen Verbindungsaufbau vom Kommunikationspartner.

Bevor auf die spezielle Parametrierung der beiden grundsätzlich zu unterscheidenden Fälle eingegangen werden soll, werden die Parameter und deren Parametrierung vorgestellt, die unabhängig von der Rolle, die die S7 Station beim Verbindungsaufbau spielt, immer gleich sind.

Der Parameter block_length enthält die Länge eines Parametrierblocks und wird fest auf den Wert 64 gesetzt. Der Parameter darf nicht verändert werden.

Der Verbindungstyp wird über den Parameter connection_type eingestellt. Hier wird fest der Wert 1 eingetragen. Dies bedeutet, dass es sich um den Verbindungstyp "TCP/IP native" handelt. Auch dieser Wert darf nicht verändert werden.

Ein weiterer Parameter, der nicht verändert werden darf und fest belegt ist, ist local_device_id. Hier ist der Wert 2 (bedeutet Industrial Ethernet) einzutragen und darf nicht verändert werden. Des Weiteren sind die beiden Parameter rem_subnet_id_len und next_staddr_len mit dem Wert 0 zu belegen und dürfen ebenfalls nicht verändert werden.

Daneben existieren noch die Parameter rem_subnet_id, next_staddr und ein reservierter Bereich, der mit spare gekennzeichnet ist. Der Parameter bzw. der reservierte Bereich ist für den Verbindungsaufbau bzw. die Kommunikation über eine TCP-/IP-Verbindung nicht relevant. Nichtsdestotrotz sollten die Parameter bzw. der reservierte Bereich nur mit dem Wert 0 beschrieben und der Wert sollte beibehalten werden.

Auf die spezielle vom Verbindungsaufbau abhängige Parametrierung soll nun im Folgenden einzeln eingegangen werden.

- zu 1:

Wird der Verbindungsaufbau aktiv von der S7 Station betrieben, so ist der Parameter active_est auf den Wert TRUE zu setzten. Somit ist festgelegt, dass der Verbindungsaufbau aktiv von der S7 Station aus betrieben wird.

Der Verbindung wird über den Parameter id eine eindeutige Nummer vergeben, über die diese Verbindung referenziert werden kann. Diese Referenz wird für die Parametrierung der Funktionsbausteine TCON, TSEND, TRCV und TDISCON benötigt.

Bei einem aktiven Verbindungsaufbau von der S7 Seite ausgehend ist die lokale Port-Nummer irrelevant. Daher wird der Parameter local_tsap_id_len auf den Wert 0 gesetzt.

Der Parameter rem_staddr_len wird auf den Wert 4 gesetzt, da eine gültige IP-Adresse sich aus vier - in der schriftlichen Darstellung durch einen Punkt getrennten - mit einem Byte darstellbaren Zahlen ausdrücken lässt.

Im Parameter rem_tsap_id_len wird die Länge der Port-Nummer des Kommunikationspartners (SIMOTION-Gerät) angegeben. Da sich die Port-Nummer mit zwei Bytes darstellen lässt, wird dieser Parameter auf 2 gesetzt.

Da die Port-Nummer auf S7 Seite in diesem Fall keine Rolle spielt und der Parameter local_tsap_id_len den Wert 0 hat, muss der Parameter local_tsap_id nicht belegt werden.

Im Parameter rem_staddr, der als Array von Byte-Größen aufgebaut ist, wird die IP-Adresse des SIMOTION-Gerätes angegeben. Dabei werden. Die ersten vier Bytes des Arrays belegt. Wobei die Stellen der IP-Adresse von rechts nach links in das Array aufsteigend eingetragen werden. D.h. die rechte Zahl wird als hexadezimale Zahl in den ersten Index des Arrays eingetragen. Die zweite Zahl von rechts wird dann in den zweiten Index eingetragen, usw.

Um die Parametrierung abzuschließen, wird in den Parameter rem_tsap_id, der ebenfalls als Array von Byte-Größen aufgebaut ist, die Port-Nummer des SIMOTION-Gerätes aufsteigend eingetragen. Im ersten Index steht das niederwertige Byte der in hexadezimales Zahlenformat gewandelten Port-Nummer. Im zweiten Index steht das höherwertige Byte der in hexadezimales Zahlenformat gewandelten Port-Nummer

- zu 2:

Wird von der S7 Station kein aktiver Verbindungsaufbau betrieben, so ist der Parameter active_est auf den Wert FALSE zu setzten. Dadurch ist festgelegt, dass von der S7 Station aus kein aktiver Verbindungsaufbau durchgeführt wird.

Auch in diesem Falle wird der Verbindung auf S7 Seite über den Parameter id eine eindeutige Nummer vergeben, über die diese Verbindung referenziert werden kann. Die eindeutige Nummer wird beim Funktionsbausteinaufruf den Funktionsbautseinen TCON, TSEND, TRCV und TDISCON übergeben.

Wie bereits oben ausgesagt lässt sich eine Port-Nummer mit zwei Bytes darstellen. Die lokale Port-Nummer auf S7-Seite ist relevant und daher muss der Parameter local_tsap_id_len auf den Wert 2 gesetzt werden.

Wie im vorhergehenden Fall a wird rem_staddr_len auf den Wert 4 gesetzt, denn eine gültige IP-Adresse setzt sich aus vier - in der schriftlichen Darstellung durch einen Punkt getrennten - mit einem Byte darstellbaren Zahlen zusammen.

Im Parameter rem_tsap_id_len wird die Länge der Port-Nummer des Kommunikationspartners (SIMOTION-Gerät) angegeben. Die Port-Nummer auf SIMOTION-Seite ist aber in diesem Fall nicht relevant. Aus diesem Grund wird der Parameter rem_tsap_id_len auf den Wert 0 gesetzt.

Im Parameter local_tsap_id - ein Array von Byte-Größen - wird die Port-Nummer der S7 Station aufsteigend eingetragen. Im ersten Index steht das niederwertige Byte der in hexadezimales Zahlenformat gewandelten Port-Nummer. Im zweiten Index steht das höherwertige Byte der in hexadezimales Zahlenformat gewandelten Port-Nummer.

Im Parameter rem_staddr (Array von Byte-Größen) wird die IP-Adresse des SIMOTION-Gerätes angegeben. Dabei werden. Die ersten vier Bytes des Arrays belegt. Wobei die Stellen der IP-Adresse von rechts nach links in das Array aufsteigend eingetragen werden. D.h. die rechte Zahl wird als hexadezimale Zahl in den ersten Index des Arrays eingetragen. Die zweite Zahl von rechts wird dann in den zweiten Index eingetragen, usw.

Da die Port-Nummer auf Seiten des SIMOTION-Gerätes in diesem Fall keine Rolle spielt und der Parameter rem_tsap_id_len den Wert 0 hat, muss der Parameter rem_tsap_id nicht belegt werden.

Es folgt die Beschreibung der Funktionsbausteine, mit denen eine Verbindung zu einem SIMOTION-Gerät auf- und abgebaut werden kann und mit denen Daten zum SIMOTION-Gerät gesendet bzw. von einem SIMOTION-Gerät empfangen werden können.

```
CALL "TCON" , DB66
     REQ       :=M1.0
     ID        :=W#16#1
     DONE      :=M2.0
     BUSY      :=M3.0
     ERROR     :=M4.0
     STATUS    :=MW100
     CONNECT   :=P#DB1.DBX0.0 BYTE 64
```

Aufruf-Beispiel des Funktionsbausteins FB65 (TCON)

Sollen Daten von einem SIMOTION-Gerät auf einer S7 Station mit integrierter Ethernetschnittstelle empfangen bzw. Daten von einer S7 Station mit integrierter Ethernetschnittstelle an ein SIMOTION-Gerät gesendet werden, so muss zuerst über den Funktionsbaustein FB65 TCON eine Verbindung zwischen der S7 Station und dem SIMOTION-Gerät aufgebaut werden.

In Programmbeispiel oben wird ein Aufuf-Beispiel des Funktionsbausteins FB65 TCON dargestellt. Über den Parameter REQ wird der Verbindungsaufbau gesteuert. Wird der Parameter auf den Wert 1 gesetzt und somit eine Flanke erzeugt, so werden die Daten (Verbindungsbeschreibung) aus dem unter CONNECT angegebenen Bereich dem Funktionsbaustein übergeben, um die Verbindung aufzubauen.

Durch den Parameter ID wird eine Referenz zur gewünschten Verbindung, die aufgebaut werden soll, hergestellt.

Über die Parameter DONE, BUSY und ERROR kann der Status der Bearbeitung des Funktionsbausteins abgefragt werden. Zusätzlich zur Information, dass ein Fehler aufgetreten ist (ERROR = 1) erhält der Benutzer noch durch den Parameter STATUS detaillierte Auskunft über die Art des Fehlers.

Wie bereits oben erwähnt, enthält der Parameter CONNECT die Adressen und Länge der Verbindungsbeschreibung. Diese Adresse weist auf einen Datenbausteinbereich hin, dessen Struktur der UDT65 entspricht.

```
Call "TSEND" , DB63
     REQ         :=M5.0
     ID          :=W#16#1
     LEN         :=10
     DONE        :=M6.0
     BUSY        :=M7.0
     ERROR       :=M8.0
     STATUS      :=MW200
     DATA        :=DB10.DBBO
```

Beispiel eines Aufrufs des Funktionsbausteins FB63 (TSEND)

Ist eine Kommunikationsverbindung aufgebaut, so können über sie Daten von der S7 Station mit integrierter Ethernetschnittstelle zum SIMOTION-Gerät gesendet werden. Dies geschieht durch den Aufruf des Funktionsbausteins FB63 TSEND.

Das Senden wird mit einer steigenden Flanke am Parameter REQ aktiviert. Beim erstmaligen Aufruf werden die Daten aus dem mit Parameter DATA angegebenen Bereich an den Funktionsbaustein übergeben.

Über den Parameter ID wird die Kommunikationsverbindung referenziert, über die die Daten versendet werden sollen. Mit dem Parameter LEN wird die Länge der zu versendenden Daten in Bytes angegeben.

Auch hier dienen die Parameter DONE, BUSY und ERROR der Angabe des Bearbeitungsstatus des Funktionsbausteins. Zusätzlich zur Information, dass ein Fehler aufgetreten ist (ERROR = 1) erhält der Benutzer durch den Parameter STATUS noch detaillierte Auskunft über die Art des Fehlers.

Der Parameter DATA enthält, wie bereits oben angegeben, die Adresse und Länge des Sendebereichs.

```
CALL "TRCV" , DB64
    EN_R      :=M8.0
    ID        :=W#16#1
    LEN       :=10
    NDR       :=M9.0
    BUSY      :=10.0
    ERROR     :=11.0
    STATUS    :=MW300
    RCVD_LEN  :=MW310
    DATA      :=DB20.DBB0
```

Beispielhafter Aufruf des Funktionsbausteins FB64 (TRCV)

Über eine aufgebaute Verbindung können auch Daten, die von einem SIMOTION-Gerät versendet werden, auf der S7 Station mit integrierter Ethernetschnittstelle empfangen werden. Hierzu wird der Funktionsbaustein FB64 TRCV aufgerufen.

Mit dem Parameter EN_R wird das Empfangen gesteuert. D.h. wird der Parameter EN_R mit dem Wert 1 beschrieben, so können Daten empfangen werden.

Über ID wird eine bestimmte Kommunikationsverbindung ausgewählt, über die die Daten empfangen werden sollen.

Für den Parameter LEN existieren zwei prinzipielle Parametrierungen. Wird der Parameter mit den Wert 0 beschrieben, so wird implizit die Länge der erwarteten Empfangsdaten über einen ANY-Pointer am Bausteineingang DATA angegeben. Sobald Daten empfangen wurden, werden die Daten im Empfangspuffer zur Verfügung gestellt und dies wird über den Parameter NDR gemeldet. Die Länge der empfangenen Daten kann dem Parameter RCVD_LEN entnommen werden und sie kann auch kleiner sein als die im Parameter DATA hinterlegte Größe. Wird der Parameter LEN mit einem Wert verschieden von 0 parametriert, so werden die empfangenen Daten im Empfangspuffer zwischen gespeichert und stehen erst dann zur Verfügung, wenn die projektierte Länge erreicht wird. Dass die Daten vollständig empfangen wurden, wird ebenfalls über den Parameter NDR gemeldet.

Der Parameter NDR meldet das teilweise oder komplette Empfangen von Daten.

Für das Empfangen geben die Parameter DONE, BUSY und ERROR den Bearbeitungsstatus des Funktionsbausteins wieder. Zusätzlich zur Information, dass ein Fehler aufgetreten ist (ERROR = 1) erhält der Benutzer durch den Parameter STATUS noch detaillierte Auskunft über die Art des Fehlers.

Die Bedeutung des Parameters RCVD_LEN wurde oben schon erwähnt. Für den Fall, dass Parameter LEN mit dem Wert 0 belegt wurde, wird im Parameter RCVD_LEN angegeben wie viele Daten in dem zuletzt empfangenen Datenblock enthalten waren. Wurde im Parameter LEN ein von 0 verschiedener Wert parametriert, so steht in RCVD_LEN derselbe Wert.

Der Parameter DATA enthält die Adresse und Länge des Sendebereichs. Dort können die empfangenen Daten zur weiteren Verarbeitung entnommen werden.

```
CALL    "TDISCON" , DB66
    REQ     :=M12.0
    ID      :=W#16#1
    DONE    :=M13.0
    BUSY    :=M14.0
```

```
ERROR   :=M15.0
STATUS  :=MW400
```

Beispielaufruf des Funktionsbausteins FB66 (TDISCON)

Der Funktionsbaustein FB66 TDISCON wird benutzt, um eine bestehende Verbindung abzubauen. Um die Verbindung abzubauen, wird der Eingangsparameter REQ auf den Wert 1 gesetzt. Der Anstoß zum Abbau der Verbindung erfolgt also durch die steigende Flanke.

Über den Parameter ID wird dem Funktionsbaustein mitgeteilt welche Verbindung abgebaut werden soll. Über diesen Parameter wird eine Referenz zu einer mittels einer Struktur vom Typ TCON_PAR definierten und bereits aufgebauten Verbindung angegeben.

Über die Parameter DONE, BUSY und ERROR kann der Status der Bearbeitung des Funktionsbausteins abgefragt werden. Zusätzlich zur Information, dass ein Fehler aufgetreten ist (ERROR = 1) erhält der Benutzer noch durch den Parameter STATUS detaillierte Auskunft über die Art des Fehlers.

6.7.4.3 SIMOTION-Funktionen

Für diesen Anwendungsfall werden auf SIMOTION-Seite dieselben Funktionen (_tcpopenclient, _tcpsend, _tcpreceive, _tcpcloseconnection, _tcpopenserver, _tcpcloseserver) verwendet, wie sie bereits beschriebenen wurden.

Allerdings werden die Port-Nummern, wie bereits erwähnt, nicht in NetPro vergeben, sondern sie werden vom Anwender in der Baustein- bzw. Funktionsparametrierung festgelegt.

6.7.5 Verarbeitung von TCP/IP-Datenpaketen im SIMOTION Anwenderprogramm

Eine Besonderheit der Kommunikation über das TCP-/IP-Protokoll ist, dass die Daten - selbst wenn die maximal übertragbare Datenlänge nicht überschritten würde - in einzelnen Paketen unterschiedlicher, nicht vorhersagbarer Größe versandt werden.

Auf S7-Seite bewirkt ein Aufruf der Funktionen FC6 (AG_RECV) oder FC60 (AG_LRECV), dass die paketweise ankommenden Nutzdaten dem Anwender in der beim Funktionsaufruf spezifizierten Länge zur Verfügung gestellt werden.

Auf SIMOTION-Seite allerdings werden dem Anwender durch den Funktions-Aufruf _tcpReceive nicht die gesamten vom Kommunikationspartner gesendeten Daten auf einmal übergeben. Sondern der Anwender muss selber dafür Sorge tragen, dass die ankommenden Datenpakete unbekannter Länge lückenlos in einen separaten Puffer und in der richtigen Reihenfolge geschrieben werden.

Im Folgenden wird in einem Flussdiagramm eine mögliche Lösung beschrieben, wie bei bekannter Gesamtdatenlänge ("Soll-Datenlänge") und unbekannter Länge der einzelnen Datenpakete (entspricht jeweils der "Ist-Datenlänge") der Empfang in SIMOTION gelöst werden kann.

Empfang auf SIMOTION-Seite mittels _tcpReceive

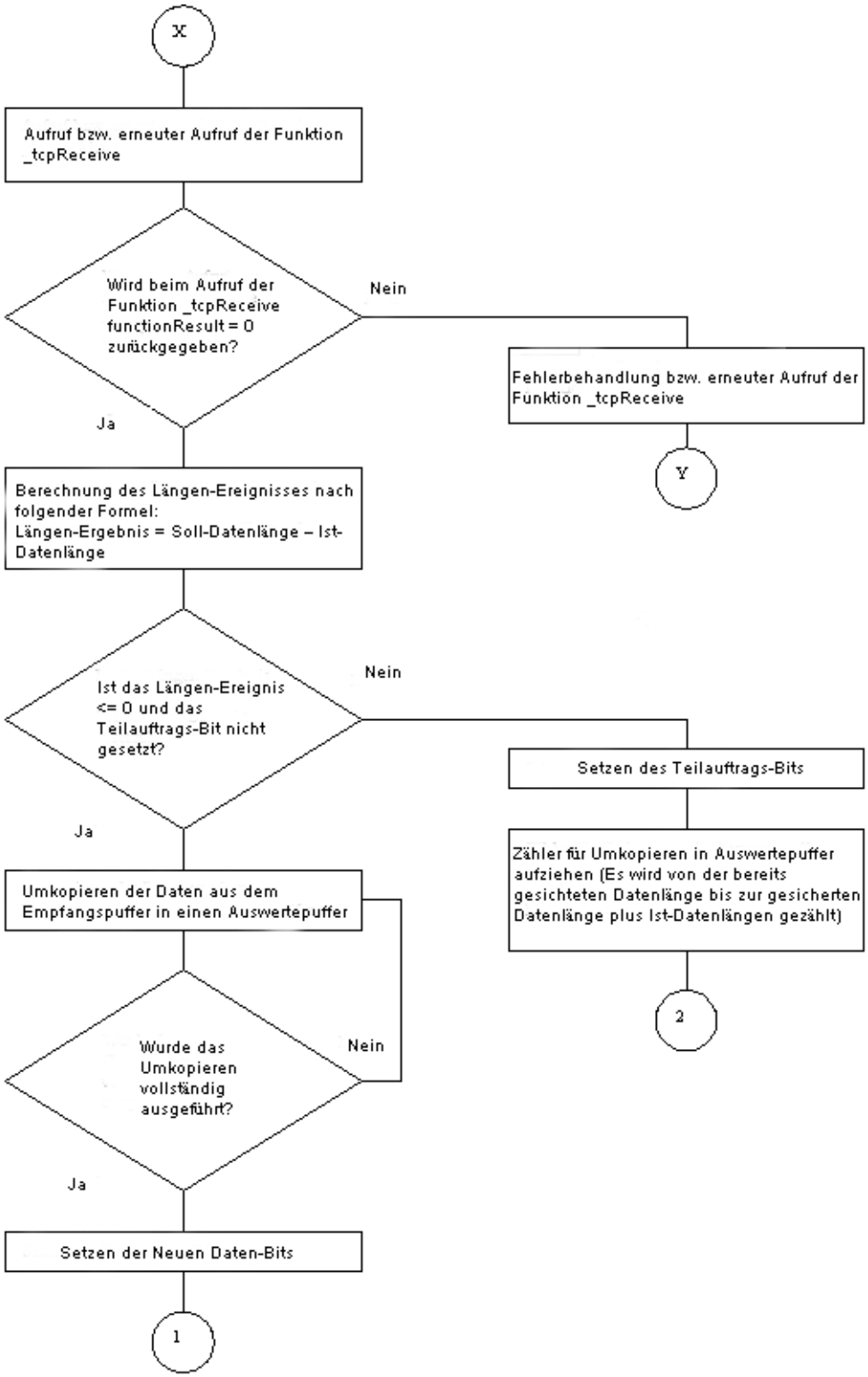


Bild 6-16 Flussdiagramm des Empfangs auf SIMOTION-Seite mittels_tcpReceive

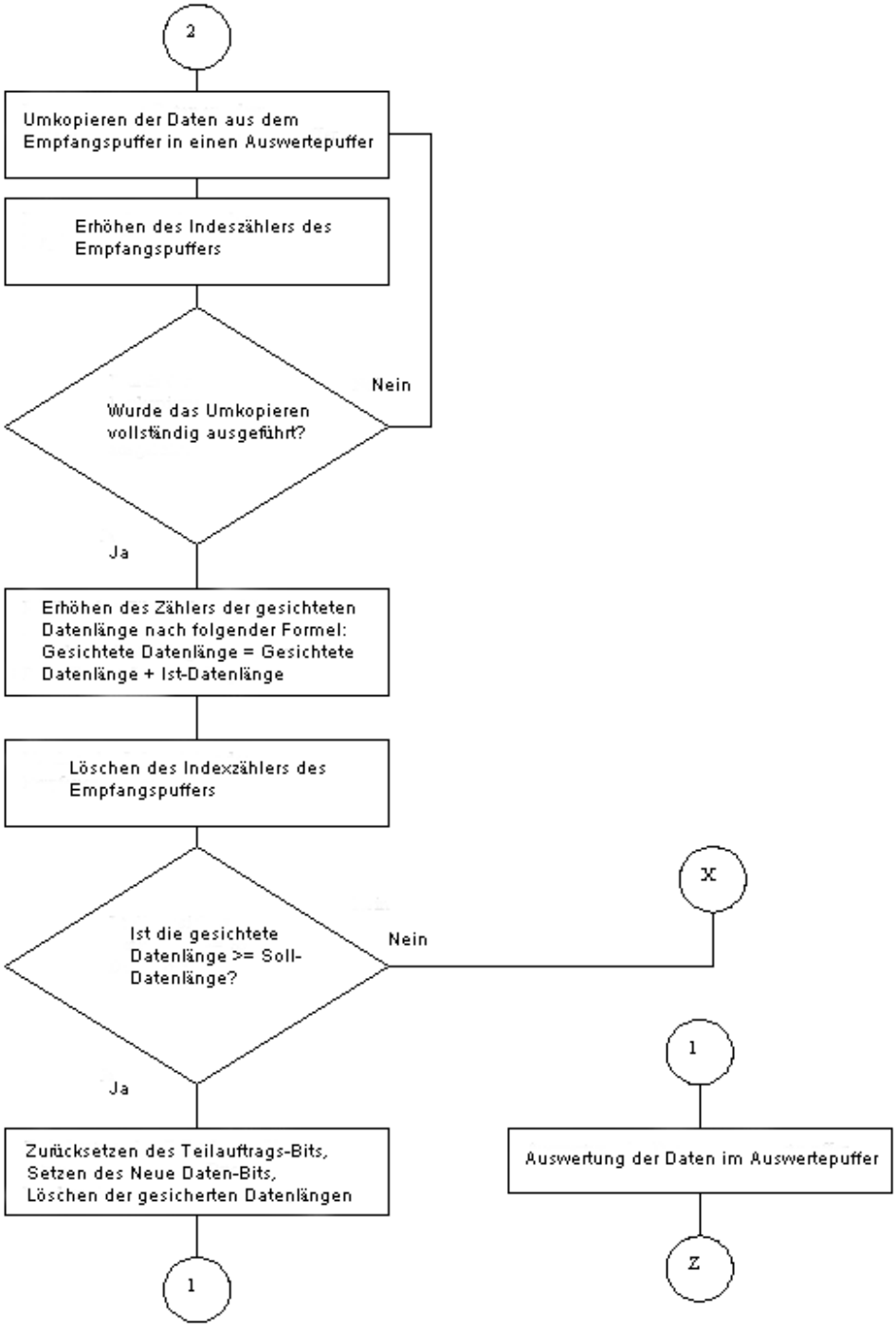


Bild 6-17 Flussdiagramm des Empfangs auf SIMOTION-Seite mittels_tcpReceive-Fortsetzung

6.8 SIMOTION TCP/IP- Systemfunktionen im Detail

6.8.1 Funktion _tcpOpenServer

Die Funktion _tcpOpenServer implementiert die Serverfunktionalität einer TCP/IP-Verbindung. Nach dem Aufruf wartet _tcpOpenServer an dem durch "port" spezifizierten Port auf die Verbindungsanforderungen von Kommunikationsteilnehmer (Clients).

Die zurückgegebene connectionId wird für nachfolgende Schreib- und Leseaufrufe (_tcpSend, _tcpReceive) benötigt.

Danach steht die Funktion für den Aufbau weiterer Verbindungen wieder zur Verfügung. Soll keine weitere Verbindung aufgebaut werden, können die durch _tcpOpenServer belegten Resourcen durch Aufruf von _tcpCloseServer wieder frei gegeben werden. Vorher aufgebaute Verbindungen werden dadurch nicht automatisch geschlossen, sondern müssen durch Aufruf von _tcpCloseConnection beendet werden.

Der Parameter "backlog" beschreibt die maximale Anzahl von Verbindungsanforderungen, die der Server während der Bearbeitung einer bereits laufenden Verbindungsanforderung zurück stauen kann.

Die Funktion muss als Rückgabewert = 16#0 haben, bevor die Datenübertragung mit TCP/IP starten kann.

Die Funktion darf nur in der BackgroundTask oder einer MotionTask aufgerufen werden.

```
_tcpOpenServer:StructRetTcpOpenServer
    (
     port :UINT,
     backlog :DINT,
     nectCommand :EnumTcpNextCommandMode
     );
```

6.8.2 Funktion _tcpOpenClient

Die Funktion _tcpOpenClient implementiert die Client-Seite einer TCP/IP-Verbindung. Nach Aufruf der Funktion erfolgt eine Verbindungsanforderung zu dem durch serverAddress und serverPort adressierten Server. Die zurückgegebene connectionId wird für nachfolgende Schreib- und Leseaufrufe (_tcpSend, _tcpReceive) benötigt.

Die Funktion muss als Rückgabewert = 16#0 haben, bevor die Datenübertragung mit TCP/IP starten kann.

Die Funktion darf nur in der BackgroundTask oder einer MotionTask aufgerufen werden.

```
_tcpOpenClient:StructRetTcpOpenClient
    (
     port             :UINT,
     serverAddress    :ARRAY [0 ... 3] of USINT;
     serverPort       :UINT;
```

```
 nextCommand      :EnumTcpNextCommandMode
);
```

6.8.3 Funktion _tcpReceive

_tcpReceive wartet an der aktiven Verbindung zum Kommunikationspartner auf Daten und liest diese aus. Die Funktion empfängt Daten über eine zuvor mit _tcpOpenServer oder _tcpOpenClient aufgebaute Verbindung. Daten können in beliebiger Stückelung empfangen werden. Die Stückelung entspricht im Allgemeinen nicht derjenigen auf der Sendeseite.

(D. h. ein Sendepaket kann in mehrere Empfangspakete aufgeteilt werden. Es können aber auch mehrere gesendete Datenpakete zu einem Empfangspaket zusammengefasst werden).

Die empfangenen Daten stehen im Parameter receiveVariable in der Länge datalength zur Verfügung, wenn im functionresult der Rückgabewert = 16#0 ist. ReceiveVariable wird beim nächsten Aufruf der Funktion mit neuen Daten in der Länge dataLength überschrieben.

Negative Werte in functionResult weisen auf einen Fehler bei der Datenübertragung hin. In diesem Fall muss die Verbindung durch Aufruf von _tcpCloseConnection abgebaut werden.

Die Funktion darf nur in der BackgroundTask oder einer MotionTask aufgerufen werden.

```
_tcp_Receive:StructRetTcpReceive
    (
     connectionId      :DINT,
     nextCommand       :EnumNextCommandMode,
     receiveVariable   :ARRAY [0 ... 4095] of BYTE
    );
```

6.8.4 Funktion _tcpSend

Mit _tcpSend werden Daten über eine zuvor mit _tcpOpenServer oder _tcpOpenClient aufgebaute Verbindung zu einem Kommunikationspartner gesendet.

Sowohl Client wie Server kann Daten über die aktive Verbindung senden.

Bei asynchronem Aufruf (nextCommand = IMMEDIATELY) muss die Funktion so lange aufgerufen werden, bis sie den Wert 0 liefert (oder einen negativen Wert im Fehlerfall).

Negative Werte in functionResult weisen auf einen Fehler bei der Datenübertragung hin. In diesem Fall muss die Verbindung durch Aufruf von _tcpCloseConnection abgebaut werden.

Die Funktion darf nur in der BackgroundTask oder einer MotionTask aufgerufen werden.

```
_tcpSend:DINT
   (
    connectionId :DINT,
    dataLength   :UDINT,
    data         :ARRAY [0 ... 4095] of BYTE
```

```
    nextCommand  :EnumTxpNextCommandMode [IMMEDIATELY | WHEN_COMMAND_DONE]
);
```

6.8.5 Funktion _tcpCloseConnection

Eine aktive Verbindung wird durch den Aufruf der Funktion _tcpCloseConnection beendet, die vorher mit _tcpOpenServer und _tcpOpenClient aufgebaut wurde und gibt damit die belegten Kommunikations-Resourcen wieder frei.

Die Funktion darf nur in der BackgroundTask oder einer MotionTask aufgerufen werden.

```
_tcpCloseConnection:DINT
    (
     connectionId:DINT
    );
```

6.8.6 Funktion _tcpCloseServer

Die Funktion _tcpCloseServer beendet den durch _tcpOpenServer eingeleiteten Wartezustand auf eine Verbindungsanforderung eines Kommunikationspartner (Client).

Die Funktion darf nur in der BackgroundTask oder einer MotionTask aufgerufen werden.

```
_tcpCloseServer:DINT
   (
    port:UINT
   );
```

6.9 SIMOTION UDP-Systemfunktionen im Detail

6.9.1 Funktion _udpSend

Beschreibung

Die Funktion _udpSend sendet ein UDP (User Datagram Protocol) Telegramm an den durch die IP-Adresse und Portnummer identifizierten Empfänger.

Folgende Daten sind für das Senden mindestens notwendig:

- IP-Adresse des Kommunikationspartners
- die "eigene" Port-Nummer
- die Port-Nummer des Kommunikationspartners

Syntax

```
_udpSend                :DINT
     (
     sourcePort          :UINT;
     destinationAddress  :ARRAY[0...3] of USINT;
     destinationPort     :UINT;
     communicationMode   :EnumUdpCommunicationMode
                          [ CLOSE_ON_EXIT |
                          DO_NOT_CLOSE_ON_EXIT ]
                          (default-Einstellung: DO_NOT_CLOSE_ON_EXIT );
       dataLength         : UDINT;
       data               : ARRAY  of Byte;
    )
```

- Der Befehl ist synchron bezüglich der Datenübergabe am Port, nicht jedoch zur Kommunikation.
- UDP ist kein sicheres Übertragungsprotokoll. Eine Rückmeldung über den Erfolg der Datenübertragung müssen Sie im Anwenderprogramm selbst programmieren.

Eine detaillierte Beschreibung der Übergabeparameter finden Sie in den Referenzlisten der SIMOTION-Systemdokumentation.

6.9.2 Funktion _udpReceive

Beschreibung

Die Funktion _udpReceive empfängt ein UDP-Telegramm an einem per Übergabeparameter spezifizierten Port.

Syntax

```
_udpReceive    :StructRetUdpReceive
    (
    port              :UINT; //( Angabe des Ports, der gelesen werden soll
)
    communicationMode :EnumUdpCommunicationMode ( vgl. _readRecord )
                       [ CLOSE_ON_EXIT |
                       DO_NOT_CLOSE_ON_EXIT ]
                       (default-Einstellung: DO_NOT_CLOSE_ON_EXIT);
     nextCommand        :EnumNextCommandMode
```

```
                    [ IMMEDIATELY |
                    WHEN_COMMAND_DONE |
                    ABORT_CURRENT_COMMAND ]
                    (default-Einstellung: IMMEDIATELY  );
 receiveVariable    :ARRAY of BYTE;
)

StructRetUdpReceive
    functionResult  :DINT;
    sourceAddress   :ARRAY[0...3] of USINT;
    sourcePort      :UINT;
    dataLength      :UDINT;
END_STRUCT;
```

Die Daten werden in der in "receiveVariable" angegebenen Variablen zurückgeliefert.

- Sie können auf die Angabe einer commandID in der Funktion verzichten, da über den Port der Status des Datentransfers abgefragt werden kann.
- Ein Aufruf der UDP-Funktionen aus der IPO-synchronen Task sollte vermieden werden, um bei nicht zu großzügig eingestellten IPO-Takten Ebenenüberläufe zu vermeiden.

Eine detaillierte Beschreibung der Übergabeparameter finden Sie in den Referenzlisten der SIMOTION-Systemdokumentation.

PROFINET IO 7

7.1 PROFINET IO Übersicht

7.1.1 PROFINET IO

Im Maschinenbau ist ein deutlicher Trend zu vermehrt dezentral konzipierten Maschinenkonzepten und mechatronischen Lösungen zu beobachten. Damit steigen die Anforderungen an die Antriebsvernetzung deutlich. Eine größere Zahl von Antrieben und kürzere Zykluszeiten, sowie die Nutzung von IT-Mechanismen, gewinnen zunehmend an Bedeutung.

Unter dem Namen PROFINET IO werden die beiden Erfolgskonzepte PROFIBUS DP und Ethernet zusammengeführt. PROFINET IO setzt auf 15 Jahre erfolgreicher Erfahrungen mit PROFIBUS DP auf und verbindet gewohntes Anwenderhandling mit der gleichzeitigen Nutzung von innovativen Konzepten der Ethernet-Technologie. Die sanfte Migration von PROFIBUS DP in die PROFINET-Welt ist dabei sichergestellt.

PROFIBUS DP ist ein Bussystem bei dem zu einem Zeitpunkt nur ein Teilnehmer auf den Bus sendend zugreifen (Halbduplex-Betrieb) darf. Bei PROFINET IO wird auf die auch bei Ethernet genutzte Switching-Technologie gesetzt. Dadurch werden alle Netzwerksegmente voneinander entkoppelt und das gleichzeitige Senden und Empfangen (Vollduplex-Betrieb) auf allen Leitungen ist möglich. Damit kann das Netz durch gleichzeitige Datenübertragung mehrerer Teilnehmer wesentlich effektiver genutzt werden. Weiterhin wurde die Bandbreite auf 100 MBit/s gesteigert.

Weitere Informationen

Detaillierte Beschreibungen zum Thema PROFINET finden Sie in dem Systemhandbuch *SIMATIC PROFINET Systembeschreibung*.

7.1.2 Applikationsmodell

Bei der Entwicklung von PROFINET IO wurde besonderer Wert auf den Investitionsschutz für Anwender und Gerätehersteller gelegt. Die Migration auf PROFINET IO erfolgt unter Beibehaltung des Applikationsmodells. Die Prozessdatensicht bleibt verglichen mit PROFIBUS DP vollständig erhalten auf:

- I/O-Daten (Zugriff auf Peripherie-Daten über logische Adressen)
- Datensätzen (Ablage von Parametern und Daten) und
- Anbindung an ein Diagnosesystem (Meldung von Diagnoseereignissen, Diagnosepuffer)

Konkret bedeutet dies, dass im Anwenderprogramm die bekannte Sicht für den Zugriff auf Prozessdaten verwendet wird. Bestehendes Programmier-Know-how kann weiterhin genutzt werden. Dies gilt auch für Geräteprofile, wie z.B. PROFIdrive, dass auch bei PROFINET IO verfügbar ist.

Auch die Engineering-Sicht bietet das gewohnte "Look and Feel". Das Engineering der dezentralen Peripherie erfolgt in gewohnter Weise mit den gleichen Tools, wie sie bei PROFIBUS bereits verwendet wurden.

7.1.3 IO-Controller

Der PROFINET IO-Controller entspricht in seiner Funktion dem Master bei PROFIBUS DP. Der IO-Controller z.B. eine SIMOTION D mit CBE30 (PROFINET IO Erweiterungskarte) tauscht zyklisch Daten mit dem ihm zugeordneten Peripheriegeräten (PROFINET IO-Devices) z.B. SINAMICS S120 aus.

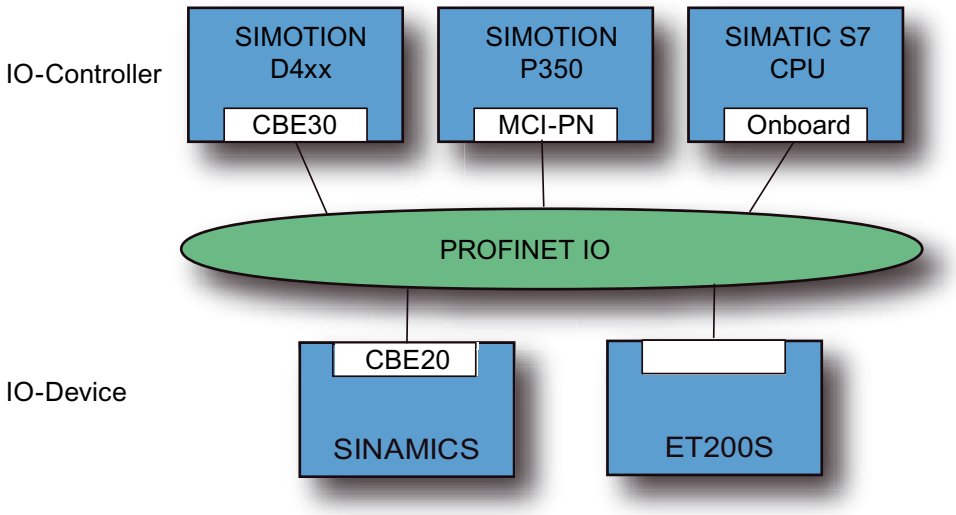


Bild 7-1 Beispiele für IO-Controller und IO-Devices

7.1.4 IO-Device

Dezentrale Feldgeräte wie I/O-Komponenten (z. B. ET200) oder Antriebe (z. B. SINAMICS S120 mit CBE20-PN) werden als IO-Device bezeichnet. Die Funktion ist mit einem PROFIBUS DP Slave vergleichbar.

Siehe auch

IO-Device anlegen (Seite 189)

7.1.5 PROFINET IO-System

Ein Controller mit dem ihm zugeordneten Devices bilden zusammen ein PROFINET IO-System.

7.1.6 Sync-Domain

Eine Sync-Domain ist eine Gruppe von PROFINET-Geräten, die auf einen gemeinsamen Takt synchronisiert sind. Der Sync-Master gibt den Takt vor. Der Sync-Slave synchronisiert sich auf den vom Sync-Master vorgegebenen Takt. Eine Sync-Domain hat einen Sync-Master.

Siehe auch

Sync-Domain anlegen (Seite 179)

7.1.7 I-Device

Das I-Device bei PROFINET ist vergleichbar mit der Funktion des i-Slaves bei PROFIBUS, d.h. eine SIMOTION CPU kann die Rolle eines IO-Device übernehmen und so mit einem anderen IO-Controller Daten austauschen.

Während bei PROFIBUS eine Schnittstelle entweder nur Master oder nur Slave sein kann, ist es bei PROFINET möglich auf einer PROFINET-Schnittstelle IO-Controller und IO-Device gleichzeitig zu sein.

7.1.8 Adressierung von PROFINET IO Geräten

Für den Datenaustausch über Ethernet wird eine weltweit eindeutige Adresse, die MAC (Media Access Control)-Adresse verwendet, die Bestandteil des Ethernet-Telegramms ist. Die MAC-Adresse ist an die Hardware gebunden und kann nicht verändert werden.

Auf Ethernet aufsetzende Protokolle wie z. B. HTTP (Webanwendungen) oder FTP (Filetransfer) nutzen das IP-Protokoll. Die Adressierung erfolgt über die IP-Adresse. Dies ist eine logische Adresse die vom Anwender vergeben werden kann.

PROFINET verwendet, neben den beiden im Zusammenhang mit Ethernet bereits bekannten Adressinformationen, zur Identifizierung der PROFINET Geräte zusätzlich einen Gerätenamen (NameOfStation). Bei dem Gerätenamen handelt es sich um einen String, der den Anforderungen eines DNS (Domain Name Service)-Namens genügt. Dieser Gerätename, auch Kommunikationsname genannt, muss eindeutig im PROFINET-Netzwerk sein.

In der Inbetriebnahme-Phase wird jedem PROFINET-Gerät (identifiziert über MAC-Adresse) durch das Projektierungswerkzeug einmalig ein Gerätename zugewiesen und dieser im PROFINET-Gerät remanent gespeichert (sog. Knotentaufe). Über den Gerätenamen wird ein Gerät in der Projektierung referenziert. Wird ein Gerät, z. B. aufgrund eines Defektes ausgetauscht, so verfügt das neue Gerät über eine andere MAC-Adresse. Wird es mit dem Gerätenamen des ausgetauschten Gerätes getauft (z. B. durch Umstecken eines Wechselmediums, das den Gerätenamen remanent speichert), kann es ohne Änderungen in der Projektierung die Funktion des getauschten Gerätes übernehmen.

Alternativ kann die Taufe der Devices anhand von Topologieinformationen automatisch vom Controller durchgeführt werden. Dazu ist es notwendig, dass im Engineeringsystem Topologieinformationen (wer ist wie mit wem verdrahtet) hinterlegt werden. Im Hochlauf identifiziert der Controller die angeschlossenen Devices über den Gerätenamen und weist dann die im Engineering definierte IP-Adresse dem Device zu. Damit ist die Station über IP-Dienste erreichbar. Die IP-Adresse kann einem projektiertem Nummerband entnommen werden oder individuell projektiert sein.

Ein PROFINET-Gerät hat folgende Adressen über das es angesprochen werden kann:

- MAC-Adresse (Bestandteil Ethernet-Telegramm, wird auf Gerät gespeichert und ist nicht veränderbar)
- IP-Adresse (IP-basierte Kommunikation z. B. Engineering-Zugriffe, muss allen Geräte zugewiesen werden)
- Gerätenamen, Kommunikationsnamen (Identifikation der Geräte im Hochlauf durch Controller)

Siehe auch

Gerätenamen und IP-Adressen für IO-Devices vergeben (Seite 195)

7.1.9 RT-Klassen

7.1.9.1 RT-Klassen bei PROFINET IO

Beschreibung

PROFINET basiert auf dem Ethernet Standard. Daher können alle auf Ethernet basierenden Standard-Protokolle (z.B. HTTP, FTP, TCP, UDP, IP ...) über das PROFINET-Netzwerk übertragen werden.

Neben den aus der Office-Welt Welt bekannten Protokollen bietet PROFINET zwei an die Anforderungen der Automatisierung angepasste Protokolle (Übertragungsarten) an. Es handelt sich um PROFINET IO mit RT und IRT.

Die beiden Übertragungsarten sind für die zyklische IO-Kommunikation mit geringen Datenmengen innerhalb eines Netzwerkes optimiert.

RT

Die RT-Kommunikation nutzt die im Ethernet Standard beschriebene Möglichkeit der Priorisierung von Telegrammen. Dieser Mechanismus wird z.B. auch bei Voice over IP genutzt. Für detaillierte Informationen, siehe PROFINET IO mit RT (Seite 146).

IRT

Bei PROFINET IO mit IRT wird dem Ethernet ein Zeitschlitzverfahren überlagert. D.h. es entstehen 2 Slots, im ersten werden die IRT-Telegramme, im zweiten die RT und IP-Telegramme übertragen. Damit wird für die IRT-Daten Übertragungsbandbreite reserviert die bei allen Last/ Überlastsituationen garantiert wird. Damit alle beteiligten Geräte wissen wann der Zeitschlitz beginnt, setzt IRT die Synchronisierung der Geräte voraus.

In der Übertragungsart IRT wird zwischen den beiden Realtime Klassen Hohe Flexibilität und Hohe Performance unterschieden.

IRT - Hohe Flexibilität

Die Realtimeklasse IRT Hohe Flexibilität entspricht der zuvor beschriebenen Übertragungsart IRT. Es wird ein für das gesamte Netzwerk einheitlicher IRT-Slot im Engineering definiert. Für detaillierte Informationen, siehe PROFINET IO mit IRT (Hohe Flexibilität) (Seite 147).

IRT - Hohe Performance

Neben der Bandbreitenreservierung erfolgt eine zeitliche Planung der zyklischen Telegramme unter Berücksichtigung der Topologie. Dadurch kann vom Engineering für jedes Kabel individuell die benötigte Bandbreite ermittelt werden. Damit kann gegenüber IRT Hohe Flexibilität das IRT-Zeitintervall minimiert und somit die Übertragung optimiert werden.

Neben der Synchronisation des Übertragungsnetzes bei IRT können bei IRT Hohe Performance auch die Applikation (z. B. Lageregler und Interpolator der SIMOTION) in den Geräten synchronisiert werden (taktsynchrone Applikation). Dies entspricht dem Verhalten des taktsynchronen PROFIBUS.

Das ist unbedingte Voraussetzung für das Schließen von Regelkreisen über das Netzwerk und taktsynchrones Schalten von Ein- und Ausgängen im Netz.

Für detaillierte Informationen, siehe PROFINET IO mit IRT (Hohe Performance) (Seite 148).

RT und IRT im Vergleich

Tabelle 7- 1 Die wichtigsten Unterschiede zwischen RT und IRT

Eigenschaft	RT	IRT (Hohe Flexibilität)	IRT (Hohe Performance)
Realtimeklasse	Realtime Class 1	Realtime Class 2	Realtime Class 3
Übertragungsart	Priorisierung der zyklischen RT-Daten durch Ethernet-Prio (VLAN-Tag)	Bandbreitenreservierung, d. h. Reservierung eines Zeitbereichs, in dem nur zyklische IRT Daten, aber keine RT oder IP-Telegramme übertragen werden.	Vom Engineering optimierte Bandbreitenreservierung auf Basis von Topologieinformationen.
Determinismus	Varianz der Übertragungsdauer der zyklischen RT-Daten durch TCP/IP-Telegramme	Garantierte Übertragung der zyklischen IRT-Daten innerhalb des reservierten IRT-Zeitintervalls	Sende- und Empfangszeitpunkte der zyklischen IRT-Daten liegen exakt fest und sind garantiert für beliebige Topologien

Eigenschaft	RT	IRT (Hohe Flexibilität)	IRT (Hohe Performance)
taktsynchrone Applikation	nicht unterstützt	nicht unterstützt	unterstützt
Hardwareunterstützung durch spezielle Ethernet-Controller	Nein	Ja	Ja

7.1.9.2 Sendetakt und Aktualisierungszeit

Beschreibung

Im PROFINET-System werden die beiden Takte Sendetakt und Aktualisierungszeit unterschieden. Der Sendetakt ist der Grundtakt für die zyklische Kommunikation. Die Aktualisierungszeit gibt an in welchem Zyklus ein Device mit Daten versorgt wird.

Sendetakt

Ist der Zeitraum zwischen zwei aufeinanderfolgenden Intervallen für IRT- bzw. RT-Kommunikation. Der Sendetakt ist das kleinstmögliche Sende-Intervall für den Datenaustausch. Der Sendetakt entspricht der kleinstmöglichen Aktualisierungszeit. Innerhalb dieser Zeit werden IRT-Daten und Nicht-IRTDaten (RT, TCP/IP) übertragen. Alle Geräte einer Sync-Domain arbeiten mit dem gleichen Sendetakt.

Aktualisierungszeit

Die Aktualisierungszeit kann für jedes IO-Device separat projektiert werden und bestimmt den Zeitabstand, in dem Daten vom IO-Controller zum IO-Device (Ausgänge) sowie Daten vom IO-Device zum IO-Controller (Eingänge) gesendet werden. Die berechneten/projektierten Aktualisierungszeiten sind immer Vielfache (2^n) des Sendetaktes.

Zusammenhang zwischen Aktualisierungszeit und Sendetakt

Die berechneten Aktualisierungszeiten sind Vielfache (1, 2, 4, 8, ..., 512) des Sendetakts. Die minimal erreichbare Aktualisierungszeit ist damit abhängig vom minimal einstellbaren Sendetakt des IO-Controllers und der Leistungsfähigkeit des IO-Controllers und des IO-Devices.

7.1.9.3 Einstellbare Sendetakte und Aktualisierungszeiten

Beschreibung

Die nachfolgende Tabelle beschreibt die bei PROFINET IO für SIMOTION-Geräte einstellbaren Sendetakte und die davon abhängig einstellbaren Untersetzungen für IRT und RT. Die einstellbaren Sendetakte sind in zwei Bereiche gegliedert: Bereich "gerade" und Bereich "ungerade". Aktualisierungszeiten ergeben sich aus dem Produkt von Untersetzungen und Sendetakt.

Tabelle 7- 2 Einstellbare Sendetakte und Aktualisierungszeiten

Sendetakt		Untersetzungen (Aktualisierungszeit = Untersetzungen * Sendetakt)	
		RT IRT Hohe Flexibilität	IRT Hohe Performance
Bereich "gerade"	250, 500, 1000 µs	1,2,4,8,16, 64,128,256,512	1 *Hinweis 2)*
	2000 µs	1,2,4,8,16,32,64,128,256	
	4000 µs	1,2,4,8,16,32,64,128	
Bereich "ungerade" *Hinweis 1*)	375, 625, 750, 875, 1125, 1250 µs ... 3875 µs (Schrittweite 125 µs)	nicht unterstützt	1

Wenn in einem PROFINET IO-System kein Sync-Master projektiert ist (kein PROFINET IRT), dann kann der Sendetakt für das betreffende PROFINET IO-System individuell am betreffenden IO-Controller in den Eigenschaften **<PROFINET-Schnittstelle>** im Reiter PROFINET unter Sendetakt oder in den Eigenschaften **PROFINET IO-System** im Reiter Aktualisierungszeit eingestellt werden. Als Default-Sendetakt ist 1 ms eingestellt. Im Register IO-Zyklus können über den Modus fixierter Faktor oder fixierte Aktualisierungszeit und den Faktor die Untersetzung für die Aktualisierungszeit eingestellt werden.

Sobald ein Sync-Master im PROFINET IO-System projektiert ist, wird der Sendetakt unter den Eigenschaften der Sync-Domain definiert. Die der Sync-Domain zugeordneten Controller übernehmen diesen Wert. Aktualisierungszeiten können für jedes IO-Device unabhängig voneinander eingestellt werden.

Hinweis 1) Mischbetrieb RT / IRT Hohe Performance
Ungerade Sendetakt können nur verwendet werden, wenn sich in den an der Sync-Domain beteiligten IO-Systemen kein RT oder IRT Hohe Flexibilität IO-Device befindet. Wenn sich IO-Devices mit RT-Klasse "RT" in einer Sync-Domain befinden, dann können nur noch die Sendetakte aus dem Bereich "gerade" eingestellt werden.

Hinweis 2) Untersetzung und taktsynchrone Applikation
Einige IO-Devices unterstützen mit IRT hohe Performance neben Untersetzung 1 weitere Untersetzungen 2, 4, 8, 16.

Wenn IO-Devices (z. B. ET200S IM151-3 PN HS, SINAMICS S) mit taktsynchroner Applikation betrieben werden, kann generell nur die Untersetzung 1 eingestellt werden. Der Modus für die Aktualisierungszeit ist dabei immer auf **fixierter Faktor** zu stellen, damit STEP 7 keine automatische Anpassung der Aktualisierungszeit vornimmt und diese damit immer dem Sendetakt entspricht. Siehe nachfolgende Abbildung.

Modus Sendetakt zur Aktualisierungszeit

- fixierter Faktor
  feste Untersetzung Sendetakt zur Aktualisierungszeit
- fixierte Aktualisierungszeit
  Aktualisierungszeit wird eingestellt
- automatisch
  STEP 7 passt die Untersetzung automatisch an, falls die Untersetzung zu klein gewählt wurde

Hinweis

Es wird empfohlen mit der Einstellung **fixierte Aktualisierungszeit** zu arbeiten.

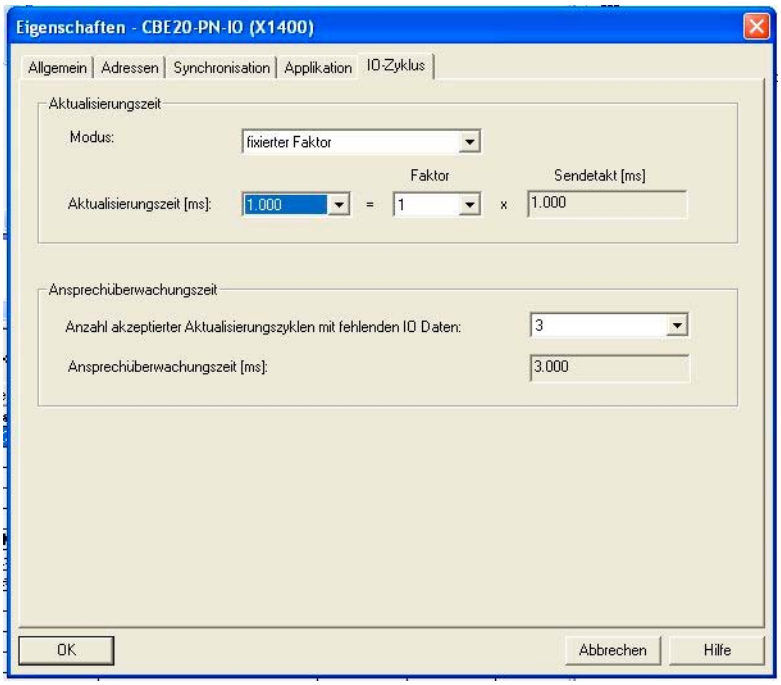


Bild 7-2 Sendetakte

7.1.9.4 RT-Klassen einstellen

RT-Klassen

Der IO-Controller bestimmt, welche RT-Klasse sein IO-System unterstützt, indem an seiner Controllerschnittstelle die Realtimeklasse eingestellt wird.

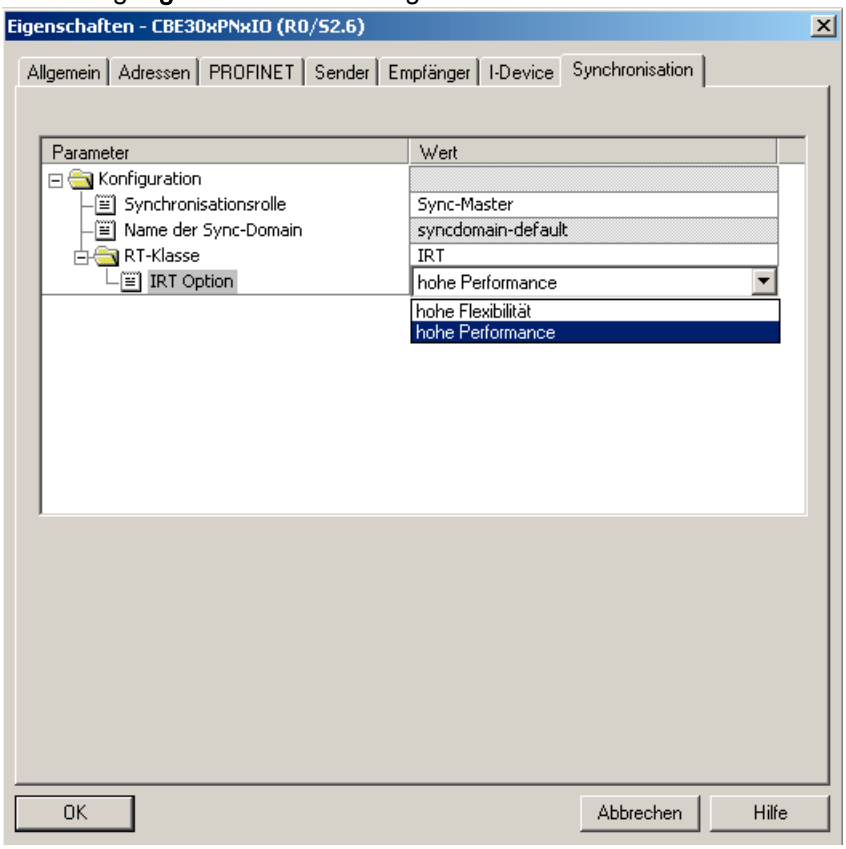
Die Realtimeklassen IRT Hohe Performance und IRT Hohe Flexibilität können in einem Netzwerk genutzt werden, schließen sich jedoch in einem IO-System aus. Ein Mischbetrieb ist **nicht** möglich. RT-Devices können immer betrieben werden, auch wenn IRT-Klassen eingestellt sind.

Einstellen der RT-Klasse

Die RT-Klasse können Sie in HW Konfig für das jeweilige PROFINET-Gerät einstellen.

1. Doppelklicken Sie in HW Konfig auf den Eintrag der PROFINET-Schnittstelle in der Baugruppe.

   Der Dialog **Eigenschaften** wird aufgerufen.



2. Wählen Sie in der Registerkarte **Synchronisation** unter RT-Klasse die Echtzeitklasse aus.
3. **Hohe Flexibilität** und **Hohe Performance** können als Option ausgewählt werden.
4. Bestätigen Sie mit **OK**.

Hinweis

Für Motion Control Applikationen mit SIMOTION und SINAMICS wird ausschließlich IRT Hohe Performance verwendet.

7.1.9.5 PROFINET IO mit RT

PROFINET IO mit RT ist die optimale Lösung für die Einbindung von Peripheriesystemen ohne besondere Anforderungen an Performance und Taktsynchronität. Es handelt sich hierbei um eine Lösung, die auf Standard-Ethernet IC (Ethernet Controller)und handelsüblichen Industrial Switches als Infrastruktur-Komponenten aufsetzt. Eine spezielle Hardware-Unterstützung ist nicht erforderlich.

Nicht taktsynchron

Standard-Ethernet und PROFINET IO mit RT bietet keine Synchronisationsmechanismen für Geräte, aber unterbinden diese Möglichkeit auch nicht. Es ist daher keine taktsynchrone Datenübertragung möglich und damit auch keine taktsynchrone Applikation für Motion Control.

Datenaustausch

Die Kommunikation über PROFINET IO mit RT und IRT basiert auf dem Ethernetframe und der MAC-Adresse. Daher ist eine Netzwerkübergreifende Kommunikation über Router hinweg mit RT und IRT nicht möglich. PROFINET IO Telegramme werden gemäß IEEE802.1Q gegenüber IT-Telegrammen priorisiert. Damit sind die in der Automatisierungstechnik erforderlichen Echtzeiteigenschaften z. B. für Standard IOs erfüllt.

Aktualisierungszeit

Die einstellbare Aktualisierungszeit liegt im Bereich von 0.25 - 512 ms. Die gewählte Aktualisierungszeit ist von den Prozessanforderungen, von der Anzahl der Geräte und der Anzahl der IO-Daten abhängig Aufgrund der Performancesteigerung bei PROFINET gegenüber Feldbussen ist der Buszyklus in der Regel nicht mehr die den Systemzyklus bestimmende Größe.

7.1.9.6 PROFINET IO mit IRT - Überblick

Überblick

Mit PROFINET IO mit IRT werden die Kommunikationsanforderungen die über Standard Signale hinaus gehen erfüllt. Mit IRT wird der bei RT noch mögliche Jitter in der Kommunikation durch die Synchronisation des Netzwerks signifikant reduziert.

Dazu wird dem Ethernet Netzwerk ein Zeitschlitzverfahren überlagert. Es wird ein Zeitschlitz für die IRT-Telegramme und einen für die RT und IP-basierten Telegramme reserviert. Voraussetzung für ein solches Verfahren ist die Synchronität aller an der IRT-Kommunikation beteiligten Geräte.

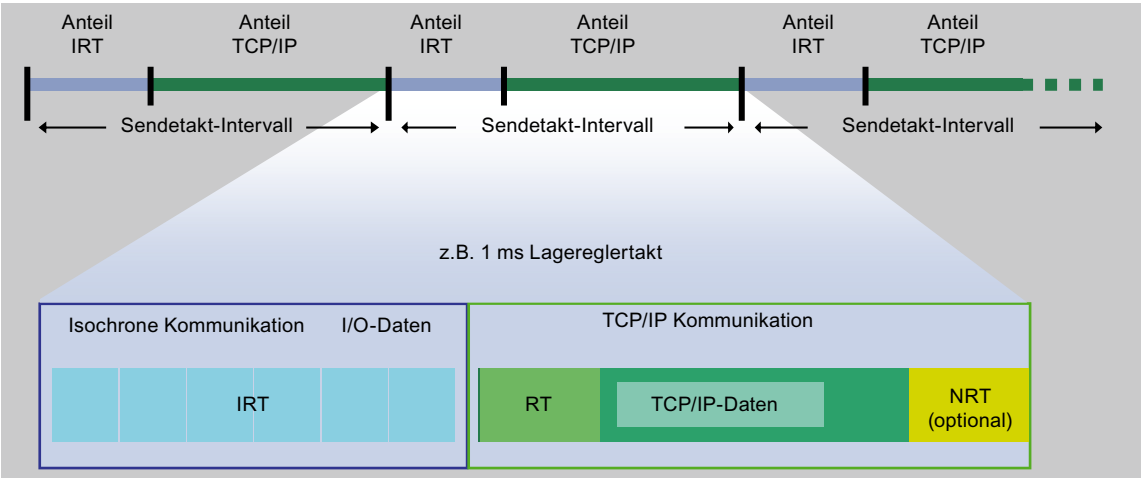


Bild 7-3 IRT Kommunikation - Überblick

PROFINET IO mit IRT gibt es in zwei Ausprägungen:

- IRT Hohe Flexibilität (Seite 147) mit fester Bandbreitenreservierung
- IRT Hohe Performance (Seite 148) mit optimierter Bandbreitenreservierung und geplanter IRT-Kommunikation

Bei PROFINET IO mit IRT werden alle IRT-Geräte auf einen gemeinsamen Sync-Master synchronisiert. Siehe auch Äquidistanz und Taktsynchronität bei PROFINET (Seite 152) .

7.1.9.7 PROFINET IO mit IRT (Hohe Flexibilität)

Beschreibung

Bei PROFINET IO mit IRT (Hohe Flexibilität) wird der höchste IRT-Bandbreitenbedarf eines Gerätes (typischerweise Controller) ermittelt und pauschal Bandbreite für das gesamte IRT-Netzwerk reserviert. Es ist die Synchronisation aller IRT-Geräte auf einen gemeinsamen Sync-Master erforderlich.

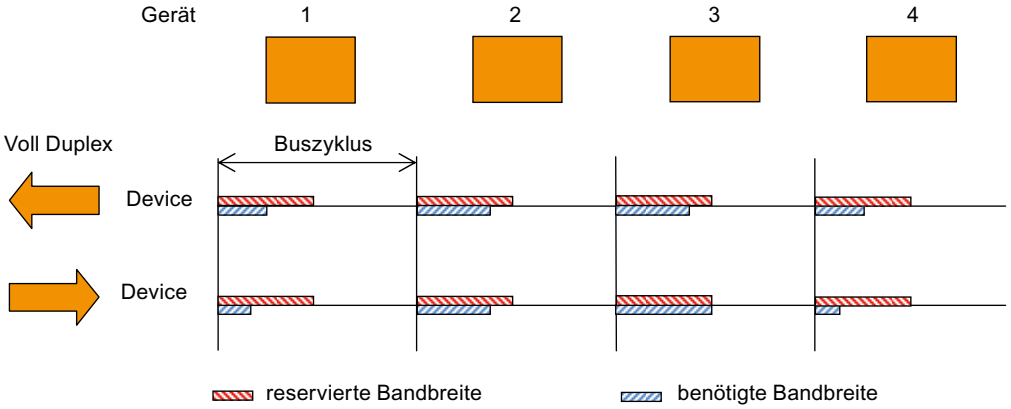


Bild 7-4 Übersicht Kommunikation mit IRT (Hohe Flexibilität)

Sendetakt

Als Sendetakt sind beim Mischbetrieb von RT und IRT Hohe Flexibilität oder auch nur IRT Hohe Flexibilität im Zusammenhang mit SIMATIC CPU nur die Werte von 0.25 (nur 319PN), 0.5 oder 1.0 möglich. Bei SIMOTION CPUs können als Sendetakt bei IRT Hohe Flexibilität die Werte 0.5 oder 1.0 eingestellt werden.

7.1.9.8 PROFINET IO mit IRT (Hohe Performance)

Für Motion Control Anwendungen wird die Leistungsfähigkeit mit PROFINET IO IRT (Hohe Performance) deutlich erweitert. Beim Einsatz von Feldbussen z.B. PROFIBUS werden die Geräte parallel an den Bus angeschlossen. Dies hat Konsequenzen, erstens es kann zu einem Zeitpunkt immer nur 1 Gerät senden. Zweitens ist aufgrund der Parallelschaltung aller Geräte mit ca. 12 Mbit/s die physikalische Grenze erreicht.

PROFINET setzt auf die Ethernet-Technologie, die Punkt zu Punkt Verbindungen vorsieht. Bei Punkt zu Punkt Verbindungen kann die Übertragungsrate gegenüber der Parallelverdrahtung erheblich gesteigert werden. Bei PROFINET wird 100 MBit/s genutzt. In Verbindung mit der Switch-Technologie werden alle Verbindungskabel voneinander entkoppelt d. h. gleichzeitig kann auf jedem Kabel gesendet und empfangen werden.

Durch die zeitliche Planung des Telegrammverkehrs bei IRT Hohe Performance erreicht man gegenüber IRT Hohe Flexibilität eine wesentliche Optimierung des Datenverkehrs, da nur die wirklich notwendige Bandbreite reserviert wird.

IRT (Hohe Performance) eignet sich besonders für:

- die Regelung und Synchronisation von Achsen über PROFINET IO
- eine schnelle, taktsynchrone Peripherieeinbindung mit kurzen Klemme-Klemme Zeiten

Sendetakt

Der Sendetakt kann bei PROFINET IRT Hohe Performance von 250 µs (P350) bis 4 ms eingestellt werden. Im Mischbetrieb RT und IRT Hohe Performance sind nur die Werte 0.5, 1.0, 2.0 oder 4.0 einstellbar.

Der tatsächlich genutzte Sendetakt ist von verschiedenen Faktoren abhängig:

- vom Prozess, es sollte nur so schnell wie nötig kommuniziert werden. Dies reduziert Buslast und CPU Belastung.
- der Busauslastung (Anzahl der Geräte und IO-Daten pro Gerät)
- in der Steuerung zur Verfügung stehende Rechenleistung
- unterstützte Sendetakte in den beteiligten PROFINET-Geräten einer Sync-Domain.

Ein typischer Sendetakt ist z.B. 1 ms; Er kann aber in einem Raster von 125 µs in den Grenzen von 250 µs bis 4 ms eingestellt werden. Siehe auch Einstellbare Sendetakte und Aktualisierungszeiten (Seite 143).

Die unterstützten Sendetakte entnehmen Sie bitte den entsprechenden Handbüchern der jeweiligen SIMOTION-Geräte. Eine minimale Zykluszeit von 250 µs wird nur von ausgewählten Komponenten unterstützt (SIMOTION P350-3 und ET 200S HS Baugruppen).

Taktsynchrone Applikation

Taktsynchrone Datenübertragung und eine auf das Bussystem synchronisierte Applikation erfüllen Anforderungen für anspruchsvolle Motion Control Anwendungen. Damit ist es möglich Regelkreise über das Bussystem zu schließen und minimale garantierte Reaktionszeiten (Klemme-Klemme-Zeitverhalten) zu erreichen. Darüber hinaus wird eine performante und taktsynchrone Anbindung an die Applikation mit geringer Belastung der Applikations-CPU sichergestellt.

Im Gegensatz zu Standard Ethernet, PROFINET IO mit RT und PROIFINET IO mit IRT Hoher Flexibilität werden die Telegramme bei PROFINET IO mit IRT Hohe Performance zeitlich geplant übertragen.

Zeitlich geplante Datenübertragung

Unter zeitlicher Planung versteht man die Festlegung der Kommunikationspfade und die exakten Übertragungszeitpunkte für die zu übertragende Daten. Durch eine Kommunikationsplanung wird die Bandbreite optimal ausgenutzt und damit eine bestmögliche Performance erzielt. Dafür muss die Topologie des Netzwerks projektiert werden und aus der projektieren Topologie berechnet das Engineering-System automatisch die Kommunikationsplanung (Siehe auch Topologie (Seite 150)). Die für PROFINET IO relevanten Daten werden durch einen Download von HW Konfig in den IO-Controller übertragen. Durch die zeitliche Festlegung der Übertragungszeitpunkte wird die höchste Determinismus-Qualität erreicht, die vor allem für eine taktsynchrone Applikationsanbindung vorteilhaft ist.

Datenaustausch

PROFINET IO mit IRT Hohe Performance und PROFINET IO mit IRT Hohe Flexibilität läuft nur innerhalb einer Sync-Domain. Diese dürfen jedoch nicht in einem IO-System gemischt werden. D. h. eine Sync-Domain kann aus 2 oder mehr IO-Systemen bestehen die alle zueinander synchronisiert werden können. Innerhalb des IO-Systems wird dann entweder IRT Hohe Flexibilität oder IRT Hohe Performance verwendet.

7.1.10 Topologiestruktur

Allgemeines

Im Folgenden erhalten Sie einen Überblick über verschiedenen Möglichkeiten, wie Sie mit SIMOTION ein PROFINET Netzwerk aufbauen können.

Tabelle 7- 3 Mögliche Topologie für SIMOTION

Topologie	
Stern	Durch den Anschluss von Kommunikationsteilnehmern an einen Switch entsteht automatisch eine sternförmige Netztopologie. Wenn ein einzelnes PROFINET-Gerät ausfällt, führt das bei dieser Struktur im Gegensatz zu anderen Strukturen nicht zwangsläufig zum Ausfall des gesamten Netzes. Lediglich der Ausfall eines Switches führt zum Ausfall der dahinter liegenden Geräte.
Baum	Wenn Sie mehrere sternförmige Strukturen miteinander verschalten, entsteht eine baumförmige Netztopologie.
Linie	Alle Kommunikationsteilnehmer werden in einer Linie hintereinander geschaltet. Wenn ein Switch ausfällt, dann ist eine Kommunikation über den ausgefallene Switch hinweg nicht mehr möglich. Für die Realisierung der Linienstruktur haben alle Geräte (außer 300 SIMATIC CPUs) einen 2 Port Switch integriert. Der Aufwand für die Verkabelung ist bei einer linienförmigen Netzstruktur am geringsten.

Topologiebeispiele Produktion

Im folgenden Beispiel sehen Sie verschiedene Topologien kombiniert.

Allgemeines zum Aufbau

PROFINET ermöglicht Ihnen Kommunikation mit hoher Performance und Durchgängigkeit. Folgende Aufbaurichtlinien sind empfohlen:

1. Schalten Sie einen Router oder einen SCALANCE S zwischen Büro-Netzwerk und PROFINET System. Über den Router können Sie den Fremdzugriff bzw. unberichtigten Zugriff auf das PROFINET-System des Produktionsnetzes verhindern.
2. Bauen Sie Ihr PROFINET System, wo sinnvoll, sternförmig auf (z. B.: verzweigen Sie nach einer CPU über ein Switch auf eine sternförmige Topologie).

3. Bei PROFINET mit IRT ist der Aufbau einer Linie mit 64 IRT-Geräten zulässig. Abhängigkeiten bestehen von den übertragenen Datenmengen. Werden größere Telegrammlängen pro Device projektiert, kann sich die mögliche Anzahl pro Linie reduzieren. Dies wird schon vorzeitig bei der Projektierung mit HW-Konfig erkannt und mit einer Fehlermeldung signalisiert. Die 64 IRT-Geräte in Linie gelten nur für PROFINET nach V2.2.
4. Halten Sie die Verkettungstiefe der Switches gering. Für die RT Kommunikation wird damit der Worst case Jitter reduziert. Bei IRT Hohe Flexibilität wird der vom Engineering ermittelte Bandbreitenbedarf reduziert.
5. Geräte mit hoher IP Last im Netwerk möglichst in einem Bereich des Netzwerks separieren. Dies sind z. B. Diagnoseserver oder HMI Server.

Beispiel-Topologie Firmennetz - Produktionsnetz

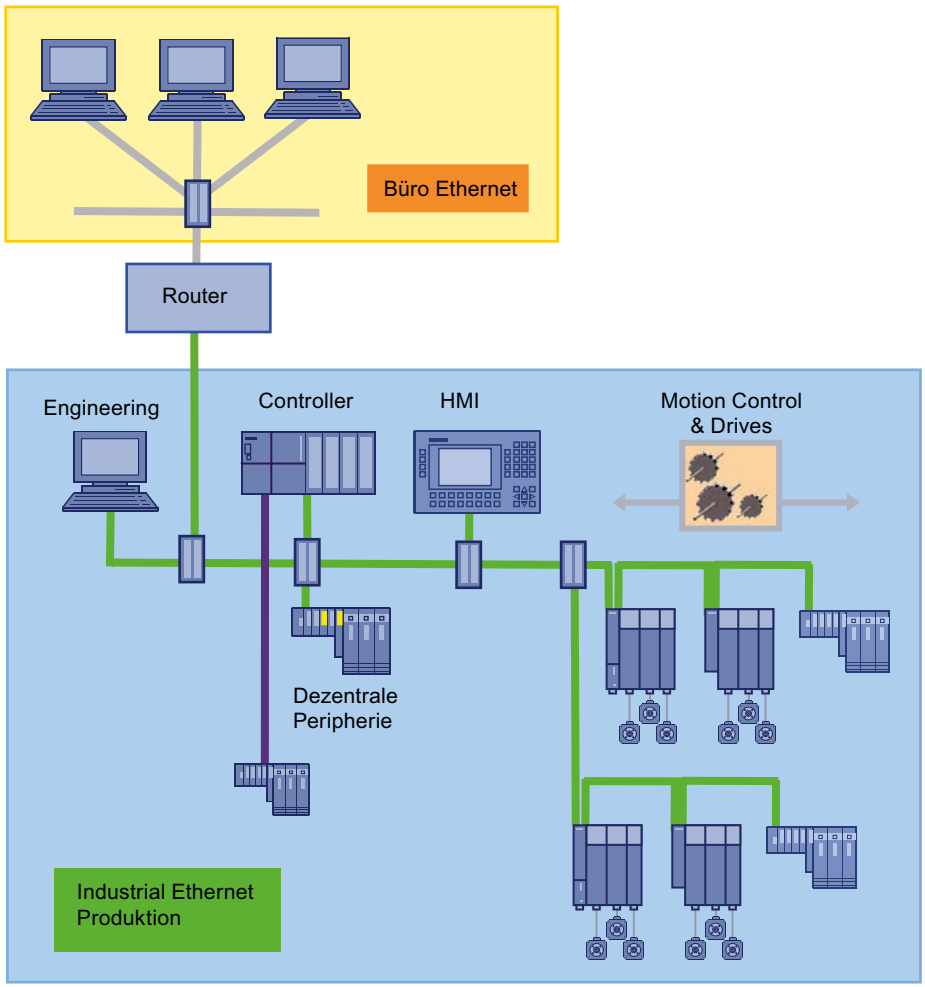


Bild 7-5 Beispiel einer Topologie

Hinweis

Weitere Informationen zur Inbetriebnahme sowie dem Topologieaufbau finden Sie in den SIMOTION Inbetriebnahmehandbüchern von SIMOTION D und SIMOTION P

Projektierung

Voraussetzung für die zeitliche Kommunikationsplanung ist die Kenntnis über die Netz-Topologie. Darunter versteht man die Informationen über das Zusammenschalten der einzelnen Geräte zu einem Kommunikationsnetz.

Die Topologieplanung ist nur für IRT Hohe Performance relevant. Die Netz-Topologie kann mit Hilfe eines in die Hardware Konfiguration integrierten Topologie-Editors benutzerfreundlich projektiert werden.

7.1.11 Äquidistanz und Taktsynchronität bei PROFINET

Beschreibung

PROFINET IO mit IRT basiert auf Ethernet mit überlagertem Zeitschlitzverfahren. Dafür ist die Synchronisation aller an der Kommunikation beteiligten Busanschaltungen Voraussetzung. Bei PROFINET IO mit IRT Hohe Performance und IRT Hohe Flexibilität überträgt ein Sync-Master ein Synchronisationstelegramm, auf das sich alle Sync-Slaves aufsynchronisieren.

Sync-Master, -Slaves und -Domain

Die Geräterollen Sync-Master und Sync-Slave werden bei der Projektierung zugewiesen. Die Rolle eines Sync-Masters kann einem IO-Controller oder SCALANCE 200 IRT-Switches zugewiesen werden.

Eine Sync-Domain kann aus PROFINET Geräten mit IRT Hohe Performance und aus PROFINET Geräten mit IRT Hohe Flexibilität bestehen. An den Enden (Stichleitungen) können sich auch PROFINET-Geräte mit RT befinden, nicht aber zwischen zwei Geräten mit PROFINET IO mit IRT (Hohe Flexibilität oder Hohe Performance).

Eine Linie (Netzwerk) darf kein IRT Hohe Flexibilität oder IRT Hohe Performance gemischt werden, denn diese müssen immer direkt miteinander verbunden sein.

Kompatibilität

Die Kommunikation zwischen und durch verschiedene Sync-Domains hindurch ist über PROFINET IO mit RT möglich.

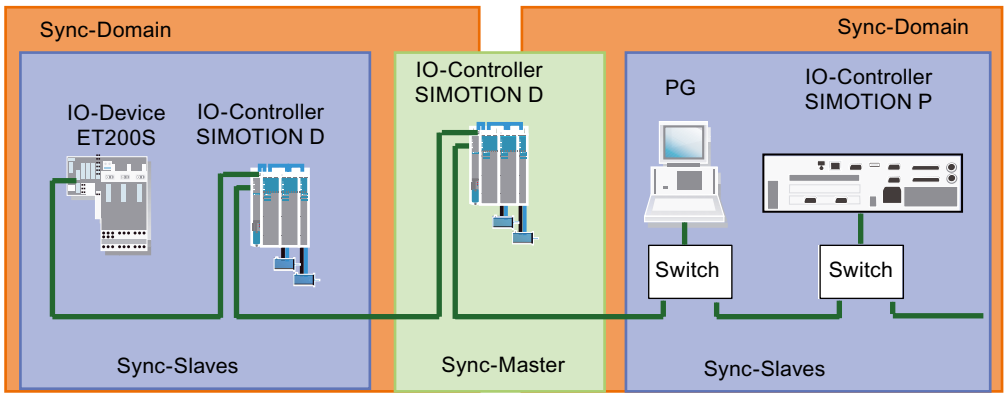


Bild 7-6 Taktsynchronität PROFINET

In dem Aufbau werden IRT-fähige Switche z.B SCALANCE X204 IRT verwendet. Alle an der IRT Kommunikation beteiligten Geräte sind direkt miteinander verbunden. Das PG in der Mitte des Netzwerkes ist über eine Stichleitung angeschlossen und unterbricht daher nicht die IRT-Strecke.

Siehe auch

Taktsynchrone Applikationen bei PROFINET (Seite 153)

7.1.12 Taktsynchrone Applikationen bei PROFINET

Wie bei PROFIBUS kann auch bei PROFINET IO mit IRT Hohe Performance die Applikation auf den Takt des Übertragungsnetzes synchronisiert werden. Die Synchronisation aller PROFINET Geräte mit IRT Hohe Performance auf eine gemeinsame Zeitbasis ist die Voraussetzung für die Realisierung verteilter Motion Control Applikationen und für Klemme/ Klemme Reaktionszeiten unter einer Millisekunde.

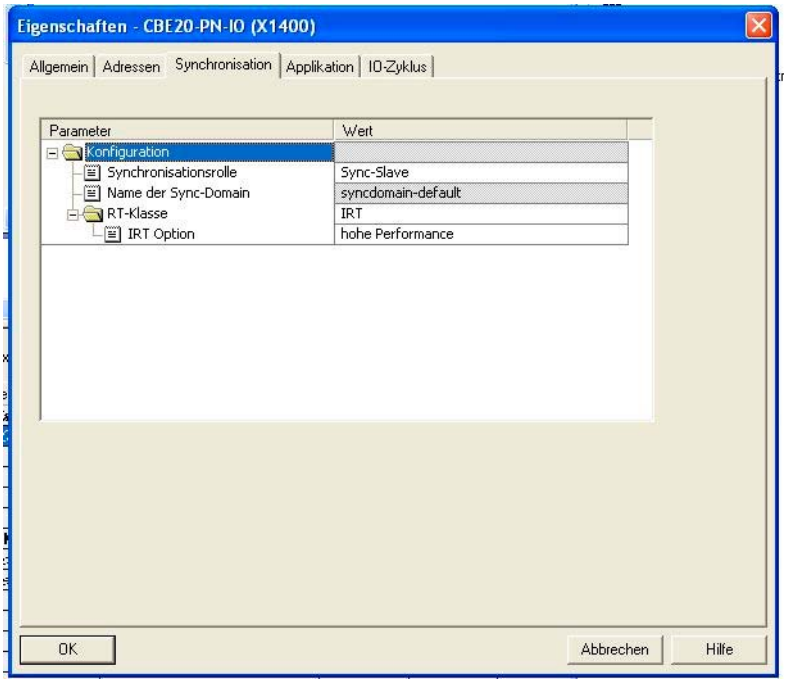
Hinweis

Die Taktsynchronität der Applikation zum Bus ist nur mit PROFINET IO mit IRT Hohe Performance möglich.

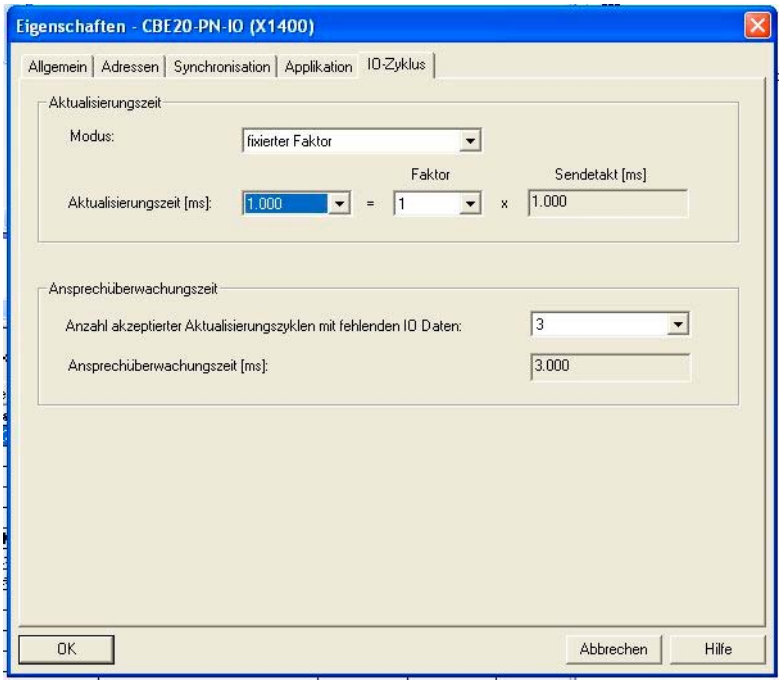
Vorgehensweise

Zur Projektierung von Taktsynchronen Applikationen gehen Sie wie im Folgenden beschrieben vor:

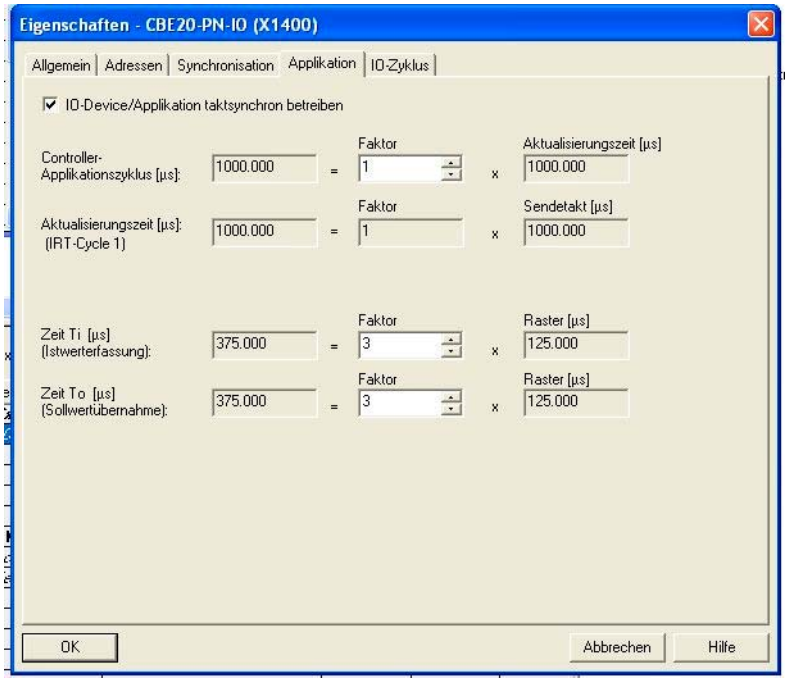
1. Stellen Sie bei Controller und Devices "IRT hohe Performance" ein.



2. Stellen Sie bei den Devices den Modus Aktualisierungszeit auf fixierter Faktor.



3. Aktivieren Sie bei den Devices die Checkbox "IO-Device/Applikation taktsynchron betreiben"



7.1.13 Taktuntersetzung

7.1.13.1 Taktuntersetzung mit PROFINET IO an SIMOTION Geräten

Beschreibung (PROFINET IO mit IRT Hohe Performance)

Eine taktsynchrone Applikation (z.B. Lageregler) auf einem IO-Controller synchronisiert sich auf den Sendetakt von IRT Hohe Performance. Sie kann sich aber auch auf ein Vielfaches des Sendetaktes der Daten synchronisieren. Dieses Vielfache wird als CACF (Controller Application Cycle Factor) bezeichnet. Die Taktuntersetzung wird wie bei PROFIBUS am Servo eingestellt (Systemtakte einstellen).

Beispiel: Die Daten auf dem Netz werden mit einem Sendetakt von 1 ms übertragen. Der Servo soll aber mit 2 msec laufen. Deshalb muss der CACF = 2 sein und am zugehörigen Antrieb eingestellt werden.

Hinweis

Taktsynchronität ist mit PROFINET IO mit IRT Hohe Flexibilität nicht möglich.

Die Einstellung des CACF erfolgt am IO-Device, siehe z.B. SINAMICS S120 einfügen und projektieren (Seite 190).

Beschreibung

Eine Untersetzung zum Sendetakt mit SIMOTION Controllern ist mit PROFINET mit IRT Hohe Performance unter folgenden Bedingungen möglich:

- Der SINAMICS Integrated einer D4xx und eine taktsynchrone DP-Master-Schnittstelle laufen immer gleich zum Servo-Takt.
- Bei einer SIMOTION P350 läuft die taktsynchrone DP-Master-Schnittstelle immer gleich zum Servo-Takt.
- Wenn ein Antrieb einer SIMOTION D, über SINAMICS Integrated, im Servotakt untersetzt zum Sendetakt läuft, darf sich der Servo-Takt über n-Sendetakte n = Untersetzung des Servos zum Sendetakt) erstrecken und muss nicht innerhalb des Sendetaktes fertig bearbeitet sein.
- Bei einem Antrieb, der nicht über den SINAMICS Integrated angeschlossen ist z. B. ein SINAMICS S120 als externes Antriebsgerät, muss der Servotakt **immer** im ersten Sendetakt gerechnet sein, d. h. es ist zwar eine Untersetzung möglich, jedoch muss der Lageregler innerhalb eines Sendetaktes gerechnet werden.

Folgende allgemeine Bedingungen gelten für Takte und Taktuntersetzungen für einen SIMOTION-Controller

- Wenn IRT Hohe Performance für ein SIMOTION-Gerät projektiert ist, aber das SIMOTION-Gerät selbst keine IRT-Daten sendet bzw. empfängt (z. B. nur Durchleiter für IRT-Daten), ist der Servotakt nicht auf den Sendetakt synchronisiert. Nur die PROFINET-Schnittstelle ist auf den Sendetakt synchronisiert, z. B.
  - Nur TCP/IP über PROFINET-Schnittstelle
  - Nur RT-Devices an der PROFINET-Schnittstelle
  - PROFINET-Schnittstelle nur *Durchleiter* für IRT Hohe Performance Daten zu anderen Geräten

  Hinweis

  Nur bei dem SIMOTION Gerät, das Sync-Master ist, synchronisiert sich das Gerät automatisch auf dem Bus auf. Sind noch weitere SIMOTION Geräte als Sync-Slaves projektiert, muss das Gerät per Applikation aufsynchronisiert werden. Dies wird in der StartupTask über den Befehl _enabelDPInterfaceSynchronizationMode programmiert.

Kombinationen von Takten und Taktquellen bei PROFIBUS und PROFINET IO

- Servo kann in ganzzahligen Vielfachen (1, 2, ...n) zum Sendetakt untersetzt werden.
- Takte des SINAMICS Integrated und taktsynchrone DP-Masterschnittstellen müssen gleich zum Servotakt laufen.

7.1.13.2 Taktuntersetzung bei Peripheriezugriffen

Beschreibung für Datenübertragung PROFINET IRT

Bei Taktuntersetzung (PROFINET und PROFIBUS) ist folgendes zu beachten:

- PROFINET IO IRT Daten werden immer zu Begin des Servo-Taktes gelesen und am Ende des Servo-Taktes geschrieben.
- PROFINET IO RT Daten werden zu Begin des IPO-Taktes gelesen und am Ende des IPO-Taktes geschrieben.
- Am Ende einer IPOSynchronousTask wird das Prozessabbild mit dem nächstmöglichen Servo (Data Out) ausgegeben (= reaktionszeitoptimiert). Dieses kann bei Untersetzung Servo-Takt zu IPO-Takt dazu führen (Servo < IPO), dass die Daten einen /mehrere Servo-Takte früher oder später innerhalb eines IPO-Taktes ausgegeben werden, wenn die Peripheriezugriffe über die IPOSynchronousTask erfolgen. Dies ist der Fall, wenn die Laufzeit des IPO-Taktes nicht konstant ist und daher bei schnellerem Servo-Takt die Daten früher oder später am Bus übertragen werden.
- Am Ende der Servo-Ablaufebene wird das Prozessabbild der ServoSynchronousTask mit dem nächstmöglichen Bus-Takt ausgegeben (= reaktionszeitoptimiert).
- Bei PROFINET und Untersetzung PROFINET-Takt zu Servo-Takt (PROFINET < Servo) kann dieses dazu führen, dass die Daten einen /mehrere Bus-Takte früher oder später innerhalb eines Servo-Taktes ausgegeben werden, wenn die Peripheriezugriffe über die ServoSynchronousTask erfolgen und die Laufzeit der Servo-Ablaufebene über einen Bustakt hinaus schwankt.
- Bei PROFIBUS werden die Daten immer mit dem ersten Bus-Takt ausgegeben, da die Servo-Ablaufebene immer mit dem ersten Bus-Takt abgeschlossen sein muss. Bei unterschiedlicher Laufzeit der Servo-Ablaufebene in den einzelnen Takten kann somit die Klemme-Klemme-Zeit variieren.

Soll statt eines reaktionszeitoptimierten Verhaltens eine immer konstante Reaktionszeit erreicht werden, so muss:

- Bei PROFIBUS:
  - ein Untersetzungsverhältnis Servo : IPO = 1 : 1 eingestellt werden, damit Peripheriezugriffe aus der IPOSynchronousTask immer taktsynchron erfolgen.
  - Anmerkung: Peripheriezugriffe aus der ServoSynchronousTask sind bei PROFIBUS immer taktsynchron
- Bei PROFINET:
  - ein Untersetzungsverhältnis Bus-Takt : Servo : IPO = 1 : 1 : 1 eingestellt werden, damit Peripheriezugriffe aus der IPOSynchronousTask immer taktsynchron erfolgen
  - ein Untersetzungsverhältnis Bus-Takt : Servo = 1 : 1 eingestellt werden, damit Peripheriezugriffe aus der ServoSynchronousTask immer taktsynchron erfolgen

7.1.13.3 Einstellbare Bustakte bei Taktuntersetzung an SIMOTION-Geräten

Übersicht über die möglichen Bustakte

	PROFIBUS	PROFINET IRT Hohe Performance	PROFINET IRT Hohe Performance	Servo
	Minimal	Minimal	Maximal	Minimal
SINAMICS S120 CU320	1 ms	0,5 ms	4,0 ms	0,5 ms
SINAMICS S120 CU310	1 ms	0,5 ms	4,0 ms	0,5 ms
C230-2	1,5 ms	-	-	1,5 ms
C240 PN	1 ms	0,5 ms	4,0 ms	0,5 ms
C240	1 ms	-	-	0,5 ms
D410 PN	-	0,5 ms	4,0 ms	2,0 ms
D410 DP	1 ms	-	-	2,0 ms
D425	1 ms	0,5 ms	4,0 ms	1,0 ms
D435	1 ms	0,5 ms	4,0 ms	1,0 ms
D445/D445-1	1 ms	0,5 ms	4,0 ms	0,5 ms
P350-3	1 ms	0,25 ms	4,0 ms	0,25 ms
ET200S HS	-	0,25 ms	4,0 ms	0,25 ms

Taktuntersetzung mit PROFINET IO

Task	Servo		IPO		IPO2	
	Min	Max	Min	Max	Min	Max
Takt	1 x Bus	16 x Bus	1 x Servo	6 x Servo	2 x IPO	64 x IPO

7.1.14 Tasksystem und Zeitverhalten

7.1.14.1 Übersicht SIMOTION Tasksystem und Systemtakte

Übersicht

Werden über PROFINET IO IRT-Daten über den Bus übertragen, sind die Takt-Laufzeiten zwischen Lesen und Schreiben der Daten z. B. Achsdaten abhängig davon in welcher Task im Ablaufsystem die Applikation ausgeführt wird. Beispiele für Applikationen in unterschiedlichen Tasks (Ablaufebenen) finden Sie in den folgenden Kapiteln.

7.1.14.2 Background-, Motion- und IPOsynchronous Task

Motion-/BackgroundTask

Die Daten werden über PROFINET IO mit IRT Hohe Performance über den Bus übertragen und vom Kommunikationsinterface zu Beginn des Servos übernommen. Die Auswertung der Logiksignale erfolgt typisch in einer Motion oder der Background Task. Dort wird entschieden welche Maschinenfunktion aktiviert wird z. B. "verfahre Achse lagegeregelt". Das notwendige Verfahrprofil wird im nächsten IPO Takt gerechnet. Aus den dort ermittelten Lagesollwerten werden im nächsten Servo-Takt die Drehzahlsollwerte für die Ansteuerung der Achse errechnet. Diese werden im nächsten Zyklus über PROFINET IO mit IRT Hohe Performance an den Antrieb übertragen.

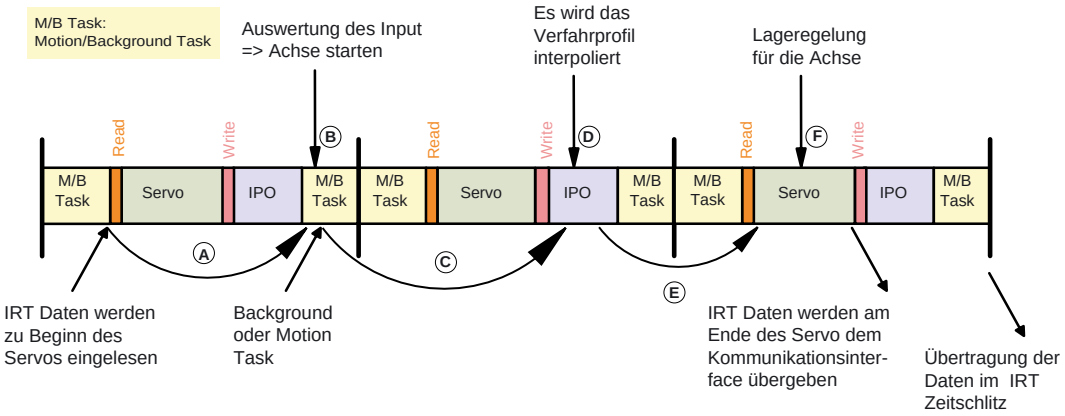


Bild 7-7 Logikauswertung für eine Achse in der Background- bzw. MotionTask

Randbedingungen

Bustakt, Servo-Takt und IPO-Takt sind im Verhältnis 1:1:1. Bei anderen Verhältnissen können sich längere Reaktionszeiten ergeben. Bei 1:1:2 kann sich die Abarbeitung des IPOs auf zwei Servo-Takte ausdehnen, damit kann sich die Reaktionszeit um einen Servo-Takt erhöhen.

Weiterhin kann sich die Bearbeitung der BackgroundTask über mehrere Servo-Takte erstrecken. Damit kann vom System nicht sichergestellt werden, dass im ersten Servo-Takt die Daten ausgewertet werden. Dadurch kann sich ebenfalls die Reaktionszeit erhöhen.

Auch die Zuordnung der Variablen zu einem Prozessabbild hat Auswirkung auf die Reaktionszeit. Die Prozessabbilder werden nicht nach Aktualisierung der Variablen sondern am Ende der jeweiligen Task den anderen Tasks bzw. dem Kommunikationsinterface zur Verfügung gestellt.

IPOsynchronousTask

Um das Zeitverhalten zu optimieren und das synchrone Auslösen von Aktionen z. B. gleichzeitiges Starten von Achsen zu ermöglichen besteht die Möglichkeit den Teil der Applikation, der Achsbefehle auslöst, in der IPOsynchronousTask zu bearbeiten. Diese wird vor dem IPO gerechnet. Dadurch kann vor der Abarbeitung des IPOs der Achsbefehl abgesetzt und im IPO dann der daraus resultierende Lagesollwert ermittelt werden. Aus diesem wird dann im nächsten Servo-Takt der Drehzahlsollwert für den Antrieb errechnet. Am Ende des Servos werden die Daten ans Kommunikationsinterface übergeben und im nächsten PROFINET IRT Sendetakt übertragen. Im Gegensatz zur Bearbeitung in MotionTask/BackgroundTask mit einer Reaktionszeit entsprechend der maximalen BackgroundTask-Laufzeit + einen IPO-Takt + einen Servo-Takt ergibt sich ein sicheres Reaktionsverhalten mit einem IPO-Takt + einen Servo-Takt bis zur neuen Datenausgabe.

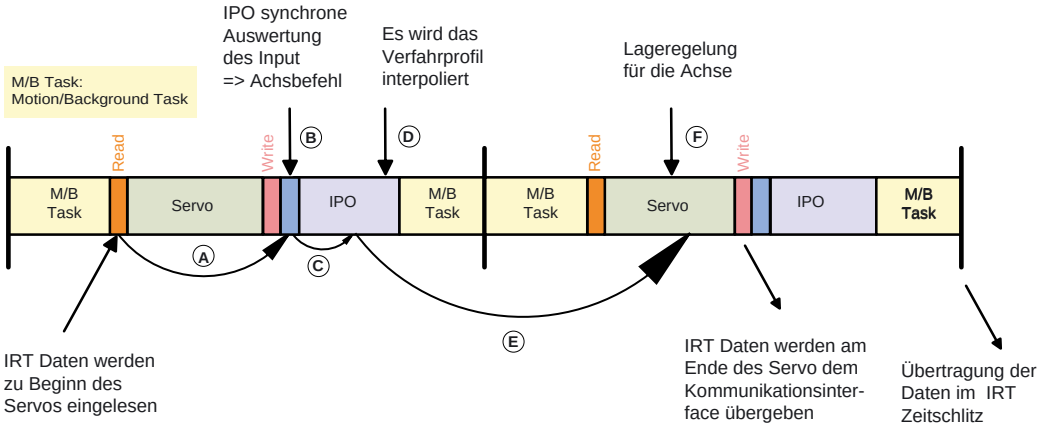


Bild 7-8 Logikauswertung für eine Achse in der IPOsynchronousTask

7.1.14.3 ServoSynchronousTask

ServoSynchronousTask

Es besteht die Möglichkeit das Zeitverhalten weiter zu optimieren und die Reaktionszeit auf einen Servotakt zu reduzieren. Dies kann für schnelle Istwertgleichläufe z. B. für das fliegende Messer/ Schere genutzt werden. Dabei wird der Teil der Applikation der die Achsbefehle für ausgewählte Achsen auslöst in der ServoSynchronous Task bearbeitet. Weiterhin wird der IPO-Systemanteil für die betroffenen Achsen in der Servo-Task vor dem Lageregler gerechnet. Dadurch können die Drehzahlsollwerte bereits im nächsten IRT-Zeitschlitz übertragen werden.

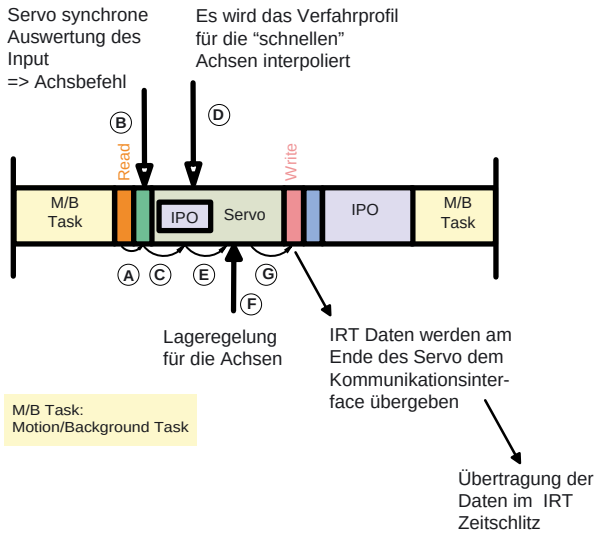


Bild 7-9 Logikauswertung für eine Achse in der ServoSynchronousTask

Randbedingungen

Die Verwendung dieser Funktion erhöht die CPU Belastung und damit den Servo-Takt. Daher sollte die Funktion nur bei Bedarf genutzt werden.

Aktivierung

Dieses Feature muss explizit für die Achsen im SIMOTION SCOUT unter der Konfiguration der Achse aktiviert werden.

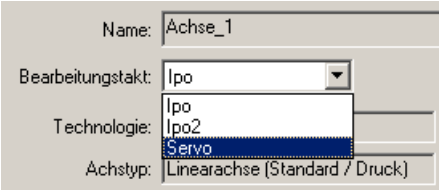


Bild 7-10 Bearbeitungstakt der Achse festlegen

Lageregelung

Die Reaktion des Lagereglers erfolgt innerhalb eines Servo-Taktes. Die Daten werden zu Beginn des Servos aus dem Kommunikationsinterface ausgelesen. Im Servo wird der Lageregler gerechnet und am Ende des Servos die neuen Drehzahlsollwerte in das Kommunikationsinterface kopiert und damit im nächsten IRT-Zeitschlitz übertragen.

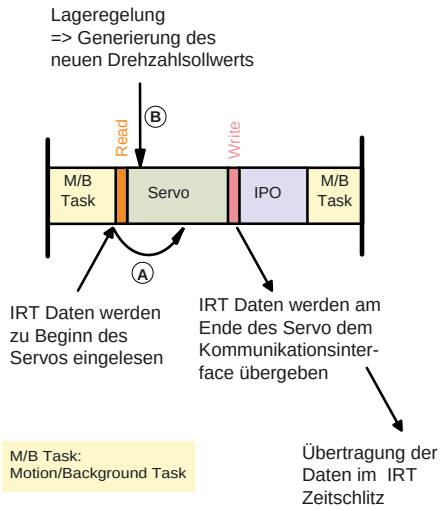


Bild 7-11 Zeitverhalten Lageregelung bei ServoSynchronousTask

7.1.14.4 Fast IO in der ServoSynchronousTask

Fast IO in ServoSynchronousTask

Die Auswertung von Fast IOs z.B. ET200S HS erfolgt in der Servo synchronen Task. Damit ergibt sich eine Reaktionszeit im System von einem Takt. Bezogen auf das Klemme-Klemme Reaktionsverhalten ergibt sich eine Verzögerung von Ti + Servotakt + To.

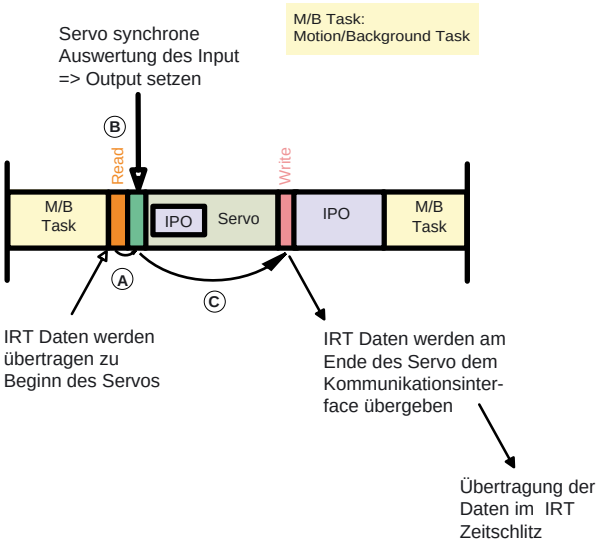


Bild 7-12 Fast IO in der ServoSynchronousTask

7.1.15 Zusammenhang Sync-Domain und IO-Systeme

Die Geräte mehrerer IO-Systeme können von einem einzigen Sync-Master synchronisiert werden, sofern sie am selben Ethernet-Subnetz angeschlossen sind und einer Sync-Domain angehören.

Umgekehrt gilt, dass ein IO-System nur einer einzigen Sync-Domain angehören kann.

7.1.16 Redundanter Sync-Master

Beschreibung

Für bestimmte Anlagen z. B. Druckmaschinen ist es erforderliche, dass Anlagenteile stand alone oder in Verbindung mit einem weiteren synchron betrieben werden können. Wenn die Gesamtanlage nur einen Sync. Master hat, wäre der jeweils andere Teil nicht für sich alleine funktionsfähig. Daher wurde die Funktion "Redundanter Sync-Master" realisiert. Dabei wird für jede Teilmaschine ein Sync-Master definiert. Einer von beiden wird als "Sync-Master" der andere als "Sync-Master (redundant)" definiert. Solange die Anlage kombiniert genutzt wird, synchronisiert sich der "Sync- Master (redundant)" auf den Sync-Master auf.

Entfällt der Sync-Master läuft der Sync-Master (redundant) autark weiter und synchronisiert die ihm zugeordneten Sync-Slaves. Die dem Sync-Master zugeordneten Sync-Slaves verlieren die Synchronisation und laufen nicht weiter.

Beschränkungen

Die beiden Sync-Master müssen direkt mit einem Kabel ohne Switch verbunden sein. Fällt die Übertragungsstrecke zwischen dem Sync-Master und Sync-Master (redundant) aus, so dass 2 Teilnetze mit je einem Sync-Master entstehen, bleiben die beiden Teilnetze auf den jeweils verbliebenen Sync-Master synchronisiert. Es entstehen damit zwei unabhängig voneinander synchronisierte Teilnetze, die u. a. durch eine Temperaturdrift der Quarze auseinanderlaufen. Nach der Wiederherstellung der Übertragungsstrecke kann sich der Sync-Master (redundant) nicht stoßfrei auf den Sync-Master aufsynchronisieren, d. h. die dem Sync-Master (redundant) zugeordneten Antriebe würden für kurze Zeit die Synchronisation verlieren und ausfallen. Für eine Synchronisation muss die Applikation gestoppt werden und neu aufsynchronisiert werden.

Zweiten Sync-Master projektieren

1. Fügen Sie eine zweite SIMOTION Baugruppe ein und projektieren Sie PROFINET entsprechend Ihren Vorgaben.
2. Klicken Sie mit der rechten Maustaste auf das PROFINET-Board um den Dialog **Eigenschaften - <PROFINET-Board> -- (R0/S2.6)** aufzurufen.
3. Wählen Sie im Register **Synchronisation** unter **Synchronisationsart** den Eintrag **Sync-Master (redundant)** aus.

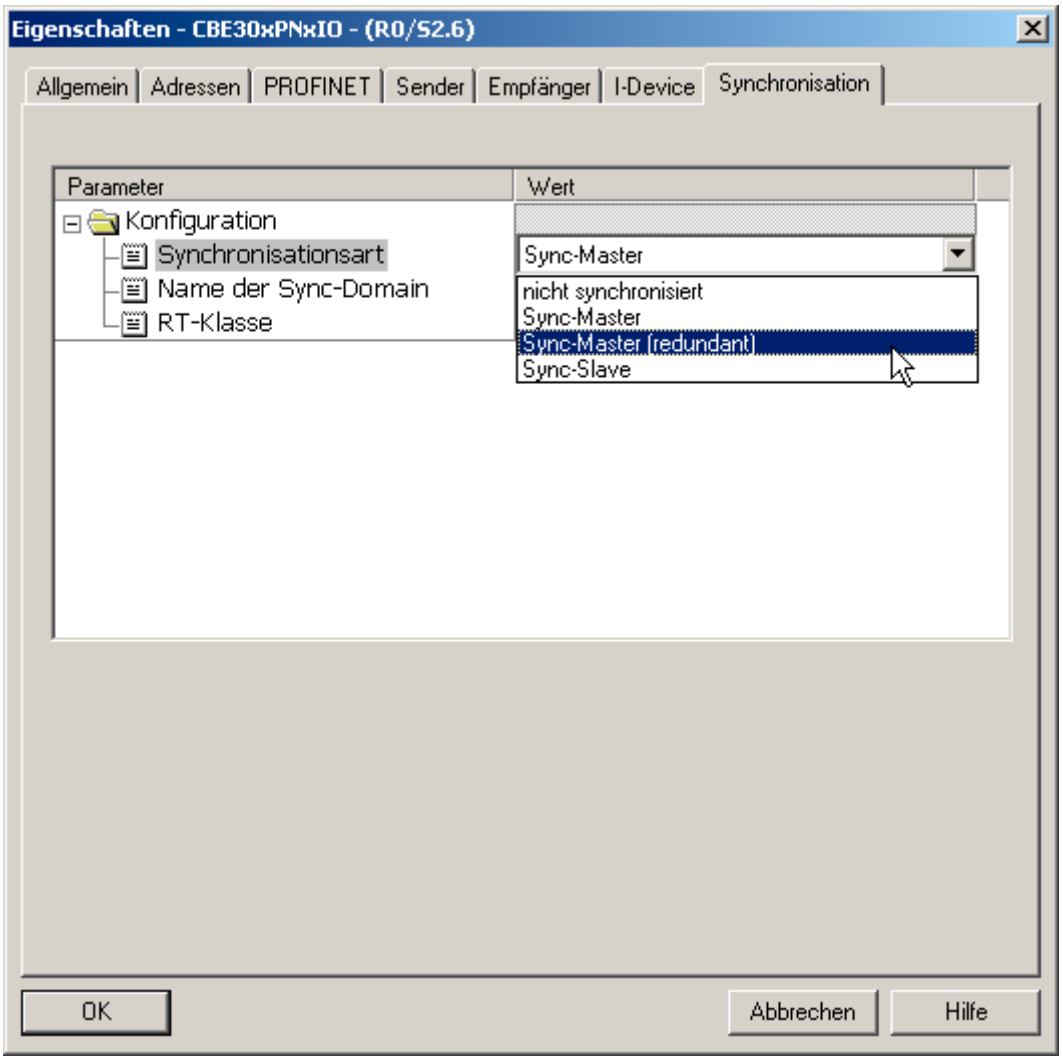


Bild 7-13 Zweiten Sync-Master projektieren

7.1.17 Mengengerüste

Für IO-Controller der SIMOTION Plattform gelten folgende Maximalwerte.

Es sind maximal 64 Kommunikationsbeziehungen möglich, diese können sich wie folgt aufteilen:

- Anschluss von maximal 64 IO-Devices.
- Maximal 64 RT bzw. RT Hohe Flexibilität Devices.
- Maximal 64 IRT Hohe Performance Devices.
- Maximal dürfen bis zu 64 Controller-Controller-Querverkehrsbeziehungen zwischen IO-Controllern aufgebaut werden.
- Wenn eine SIMOTION als I-Device projektiert wurde, zählt diese als Device mit, d.h. es sind dann nur noch 63 Devices anschließbar.
- Eine SIMOTION kann Daten von bis zu 64 anderen SIMOTION empfangen, kann aber an beliebig viele SIMOTION Daten senden.
- Beim Controller-Controller-Querverkehr ist noch die Datenmenge zu berücksichtigen. Ein Querverkehr wird nur bis zu einer bestimmten Datenmenge als eine Verbindung gezählt. Wenn ein zweites Telegramm benötigt wird, entfallen auf den Querverkehr zwei Verbindungen.

Mischbetrieb von IO-Devices und Controller-Controller Querverkehr

Die mögliche Anzahl von Geräten im Mischbetrieb können Sie sich über folgende Formeln berechnen:

IRT Hohe Performance

```
RT + IRT Hohe Performance-IO-Device + Querverkehrsframes <= 64
```

IRT Hohe Flexibilität

```
RT + IRT Hohe Performance-IO-Device <= 64
```

Hinweis

Bei einer Querverkehrsbeziehung ist nicht die Anzahl der projektierten Slots (Zeilen in den Laschen Empfänger) (siehe Empfänger projektieren (Seite 201)) für die IRT Hohe Performance Querverkehrsprojektierung gemeint, sondern die Anzahl der Ethernet-Frames, die für den Querverkehr empfangen werden. Ein Ethernetframe kann max. 768 Byte Querverkehrsnutzdaten enthalten.

Ein Slot hat maximal 254 Bytes und im Querverkehr können 3072 Bytes Nutzdaten (aufgeteilt auf 4 Frames je 768 Byte) ausgetauscht werden. Der Wert ist abhängig von den gewählten Slotgrößen. Es können daher mehrere Slots von einem Sender empfangen werden können, die aber in einem Telegramm übertragen werden können.

Jeder Provider sendet seine Querverkehrs-Daten in einem Ethernet-Frame. Jede andere SIMOTION kann diese Daten in diesen Frame lesen. So haben wir eine zählende Verbindung zu jedem sendenden SIMOTION.

Beim Controller-Device Verkehr hat ein Frame die Nutzdatengröße von 1440 Bytes.

HW Konfig prüft gemäß den oben genannten Formeln das projektierte Mengengerüst beim Übersetzen des Projektes.

Adressraum

Im logischen Adressraum eines IO-Controllers dürfen jeweils für Input- und für Output-Daten maximal 4 Kilobyte für PROFINET IO Daten belegt werden. Der Rest des insgesamt 16 Kilobyte großen Adressraumes kann z.B. durch PROFIBUS Daten oder Diagnosedaten belegt werden.

7.1.18 Azyklische Kommunikation über PROFINET

Beschreibung

Analog zu PROFIBUS DP ist es auch für PROFINET IO möglich, azyklischen Kommunikation (Base Mode Parameter Access) zu betreiben. Eine ausführliche Beschreibung dazu finden Sie unter DP-V1 Kommunikation (Seite 37).

7.2 Spezifische Eigenschaften von PROFINET IO mit SIMOTION

7.2.1 Einleitung

Voraussetzung

Um mit SIMOTION über PROFINET IO arbeiten zu können, muss entweder das Option Board CBE30 in den Option Slot der SIMOTION D4x5-Geräte gesteckt sein, eine SIMOTION P350 PN, SIMOTION D410 PN oder C240 PN verwendet werden.

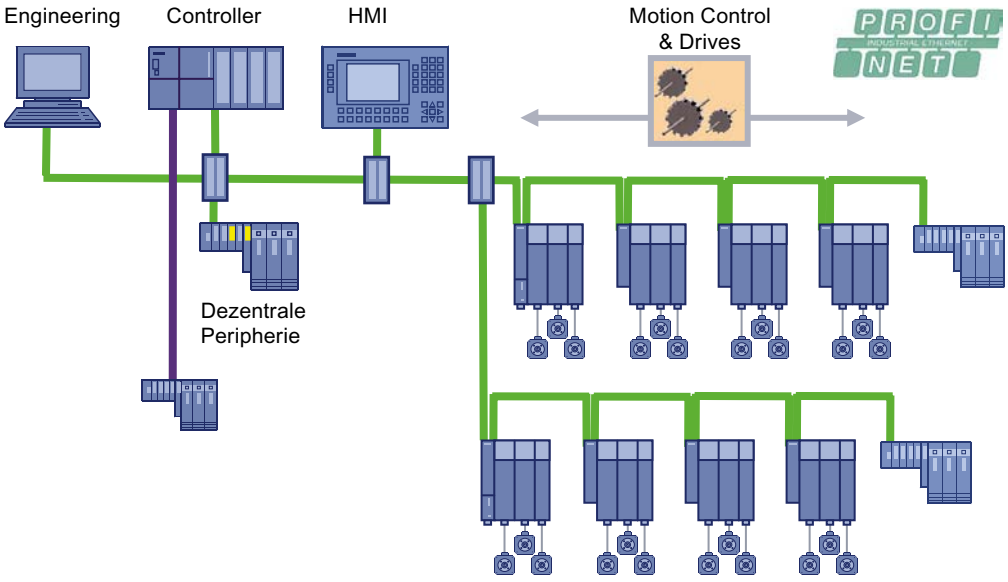


Bild 7-14 Anlagentopologie mit PROFINET

Die PROFINET-Geräte unterstützen den gleichzeitigen Betrieb von:

- IRT Hohe Performance - Isochronous Realtime Ethernet
  - Betrieb von IRT-Peripherie (z. B. ET200S HS für IRT Hohe Performance)
  - Betrieb eines SINAMICS S120 als IO-Device
  - Datenaustausch zwischen Controllern über IRT Hohe Performance (z. B. Verteilter Gleichlauf)
- RT - Realtime Ethernet
  - Betrieb von RT - Peripherie (z. B. ET 200S, ET 200pro)
  - ASi-Link über IE/ AS-Interface Link PN IO für den Netzübergang PROFINET IO auf AS-Interface
  - SINAMICS als PROFINET IO mit RT Gerät
- TCP/IP, UDP, HTTP, … Standard Ethernet Dienste

Hinweis

Bei Mischbetrieb von IRT Hohe Performance und RT bzw. IRT Hohe Flexibilität und RT ist darauf zu achten, dass die IRT (Hohe Performance oder Hohe Flexibilität)-fähigen Geräte direkt miteinander verbunden sind. D.h. es darf kein nicht IRT-fähiges Gerät zwischen die IRT-Geräte geschaltet werden.

Hinweis

Mit SIMOTION SCOUT können Sie maximal zu 10 PROFINET Teilnehmern gleichzeitig ONLINE gehen. Wenn Sie SIMATIC NET installiert haben, ist es möglich mehr als 10 Verbindungen aufzubauen.

7.3 PROFINET IO mit SIMOTION projektieren

7.3.1 Neues zu SIMOTION V4.1.2

Beschreibung

Ab SIMOTION V4.1.2 wird ein neues Synchronisationsverfahren unterstützt. Dies ist in der PROFINET Spezifikation V2.2 definiert. Das bisher als IRT top gekennzeichnete PROFINET IRT wird ab STEP7 5.4 SP4 und SCOUT V4.1.2 als **IRT*** in HW Konfig gekennzeichnet.

PROFINET V2.1

IRT top (nun IRT*) bis STEP7 5.4 SP3.1 und SCOUT < V4.1.2

PROFINET V2.2

IRT Hohe Performance ab STEP7 5.4 SP4 und SCOUT V4.1.2

IRT Hohe Flexibilität ab STEP7 5.4 SP4 und SCOUT V4.1.2

Hinweis

Alle IRT-Teilnehmer müssen entweder der Norm PROFINET V2.1 oder PROFINET V2.2 entsprechen. Mischkonfigurationen sind nicht zulässig. RT-Teilnehmer sind von der Umstellung auf PROFINET V2.2 nicht betroffen.

Standardmäßig wird die neue Hardware als PROFINET V2.2 ausgeliefert. Sollten Sie bestehende Anlagen hochrüsten bzw. neue Hardware rückrüsten, kontaktieren Sie bitte den Siemens Support.

Ab PROFINET V2.2 stehen Ihnen dann die beiden IRT-Varianten IRT Hohe Performance und IRT Hohe Flexibilität zur Verfügung, das alte IRT* (PROFINET V2.1) wird weiter unterstützt.

Übersicht der Projektierung bei PROFINET V2.1 und V2.2 mit STEP7 5.4 SP4 und SCOUT V4.1.2

Tabelle 7- 4 PROFINET Projektierung

Projektierungsschritt	IRT nach PN V2.1	IRT nach PN V2.2
Konfiguration der RT-Klasse in HW Konfig	IRT*	IRT mit Option: • Hohe Performance • Hohe Flexibilität
Konfiguration von SINAMICS S120 über GSD	GSD V2.0, GSD V2.1, GSD V2.2	GSD V2.2
Konfiguration von SINAMICS S120 über DeviceOM	Wird unterstützt	Wird unterstützt
SIMOTION	-	Ab 4.1.2
SINAMICS	-	Ab 2.5.1.10
SCALANCE X200 IRT Switch	FW 3.1.x	Ab V4.1

7.3.2 PROFINET V2.2 und PROFINET V2.1

PROFINET V2.1 und V2.2

Bei der Normung von PROFINET für die IEC61158 gab es einen Übergang von Version V2.1 nach V2.2. Dabei haben sich Inhalte bezüglich PROFINET mit IRT geändert.

Bei der Verwendung der RT-Klasse IRT im PROFINET-Netzwerk sind die Versionen V2.1 und V2.2 nicht kompatibel. D. h. bei Verwendung von PROFINET mit IRT ist bei SIMOTION und SINAMICS darauf zu achten, dass die beteiligten Geräte entweder alle die Norm nach V2.1 oder V2.2 unterstützen, sofern sie an der IRT-Kommunikation teilnehmen. Die Kommunikation mit der Realtime-Klasse RT bleibt zwischen beiden PROFINET Versionen kompatibel. Alle Siemensgeräte in aktueller Version unterstützen PROFINET V2.2.

Ab folgenden Versionen unterstützen die Geräte PNV2.2 als Defaulteinstellung:

- SIMOTION D4xx, P350-3, C240 PN >= V4.1.2
- SINAMICS S120 >= 2.6.1
- SCALANCE X200IRT >= V4.1
- ET200 HS >= V2.0
- SIMATIC NET CP1616/1604 >= 2.3

Die Liste enthält die Geräte, die IRT Hohe Performance unterstützen und deren vorherige Versionen bzgl. IRT nicht mehr kompatibel sind.

Ab folgender Version unterstützen die Engineering Tools PROFINET V2.2:

- SIMOTION SCOUT/Starter >=V4.1.2
- SIMATIC Step 7 >= V5.4 SP4

Sollte es in Maschinenkonfiguration vorkommen, dass IRT-Kommunikation mit Geräten notwendig ist, die nicht PROFINET nach V2.2 unterstützen, können SIMOTION und SINAMICS auf PROFINET V2.1 umgestellt werden. Default ist jeweils V2.2. Um SIMOTION auf V2.1 zurückzusetzen muss die entsprechende Firmware in das Gerät geladen werden. Die entsprechenden Firmwareversionen für SIMOTION D4xx befinden sich auf der SCOUT DVD unter Firmware\3_D4xx\Firmware_D4xx\V4.1.2.3\PN_V2.1and_DP\d4xx_pn21.zip. Sollte eine Rückrüstung bei SIMOTION P350 notwendig werden, wenden Sie sich an Ihren Siemens Ansprechpartner. Bei SINAMICS S120 ist das durch Setzen eines Parameters p8835=0 möglich. SIMOTION C240 PN unterstützt nur PROFINET V2.2 und kann nicht auf PROFINET V2.1 zurückgerüstet werden.

Hinweis

Wenn möglich, sollen alle Projekte mit oben genannten Versionen mit PROFINET V2.2 durchgeführt werden, da SIMOTION und SINAMICS in späteren Versionen PROFINET mit IRT nach der Version V2.1 nicht mehr unterstützen wird.

7.3.3 Vorgehensweise zur Projektierung von PROFINET IO mit IRT Hohe Performance

Vorgehensweise

Für die Projektierung von PROFINET IO mit IRT Hohe Performance V2.2 müssen Sie Folgendes durchführen:

1. Einfügen der SIMOTION Baugruppe.
   Sie wählen zuerst den **CPU-Typ** und dann die **Variante**. Beachten Sie, dass Sie die Variante mit PROFINET V2.2 wählen.
   - Die CPUs wählen Sie im Einfügedialog in SIMOTION SCOUT z. B. CPU-Typ D445-1 und Variante D445 PN-V2.2, SINAMCIS S120 Integrated V2.5.
   - Für SIMOTION D4x5 müssen Sie noch zusätzlich die CBE30-Baugruppe per Drag&Drop aus dem Hardware-Katalog in HW Konfig auf die entsprechende Schnittstelle der SIMOTION Baugruppe ziehen.
     Die CBE30-Baugruppen finden Sie im Hardwarekatalog unter **SIMOTION Drive Based > SIMOTION D4xx > 6AUxx > V4.1 - PN-V2.2xx**.
2. IO-Devices einfügen:
   Fügen Sie IO-Devices aus dem Hardware-Katalog von HW Konfig in das I/O-System ein. Die IO-Devices finden Sie im Hardwarekatalog unter **PROFINET IO**.
3. Sync-Domain projektieren und Sendetakt festlegen:
   Legen Sie fest, welche PROFINET IO Teilnehmer der Sync-Master (Taktgeber) ist und definieren Sie die Sync-Slaves. Legen Sie den Sendetakt fest.
4. Topologie erstellen:
   Legen Sie die Topologie fest, d. h. wie die einzelnen Ports der PROFINET IO Geräte untereinander verschaltet sind. Die Projektierung der Topologie ist nur für PROFINET IO mit IRT Hohe Performance notwendig. Wenn die Topologiebasierte Taufe genutzt werden soll, ist die Topologieprojektierung für alle PROFINET Geräte erforderlich
5. Controller - Controller Querverkehr:
   Legen Sie fest, welche Adressbereiche zum Senden und welche zum Empfangen verwendet werden sollen.

7.3.4 SIMOTION CPU D4x5 einfügen und projektieren

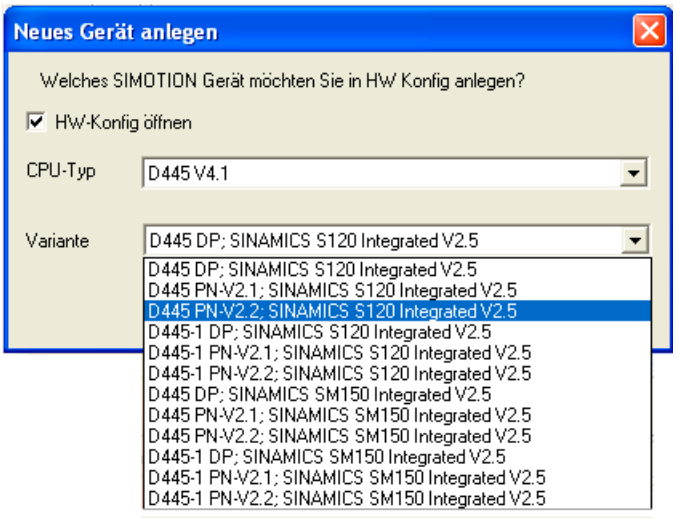
Allgemeines

Sie haben ein Projekt angelegt und wollen eine SIMOTION D4x5 unter der Verwendung einer PROFINET-Schnittstelle projektieren. Die PROFINET-Schnittstelle wird bei D4x5 über ein CBE30 realisiert. Bei D410 PN ist die PROFINET-Schnittstelle schon integriert.

Vorgehensweise

1. Klicken Sie auf neues Gerät anlegen im Projektnavigator, damit der Geräte-Auswahldialog geöffnet wird.
2. Wählen Sie den **CPU-Typ** z. B. D445 V4.1. Alle CPU der Version D4x5 V4.1 unterstützen PROFINET V2.2. Sie können aber auch Variante PN-V2.1 für PROFINET V2.1 wählen. Unter **Variante** wählen Sie die PROFINET- und Antriebsversion. Für PROFINET V2.2

müssen Sie eine Variante mit PN-V2.2 wählen z. B. D445 PN-V2.2 SINAMICS Integrated V2.5.



3. Bestätigen Sie mit **OK**. HW Konfig wird geöffnet.
4. In HW Konfig müssen Sie ein CBE30 für das entsprechende Antriebsgerät einfügen (siehe CBE30-PROFINET-Board einfügen und projektieren (Seite 173)).

7.3.5 CBE30-PROFINET-Board einfügen und projektieren

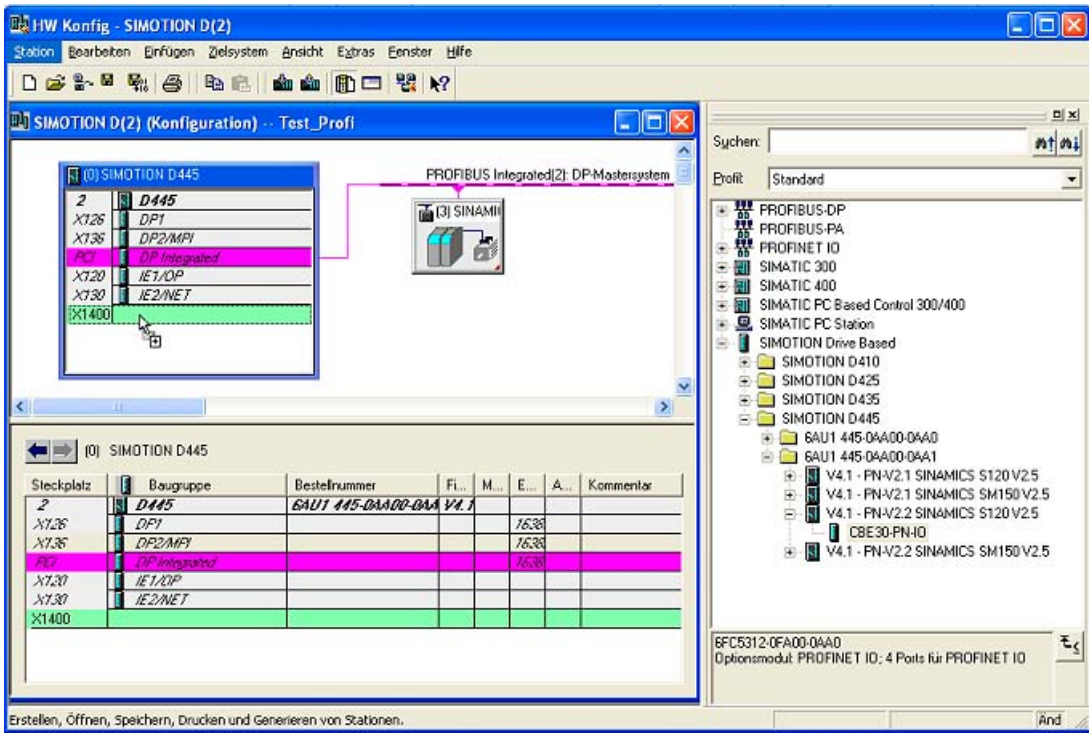
Voraussetzung

Sie haben bereits ein Projekt angelegt und ein SIMOTION-Gerät eingefügt.

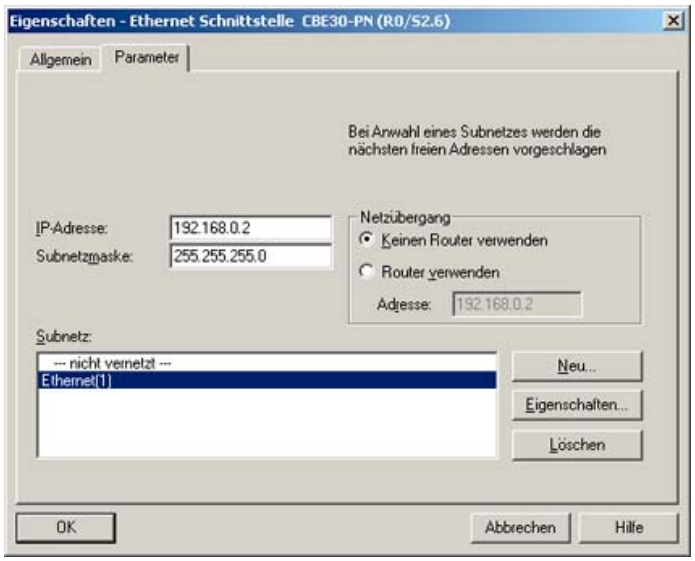
Vorgehensweise

1. Doppelklicken Sie im Projektnavigator auf die Baugruppe (hier eine D445). HW Konfig wird mit der entsprechenden Baugruppe eingeblendet.
2. Klicken Sie im Hardware-Katalog den Eintrag der Baugruppe, z.B. SIMOTION D445 an.

3. Klicken Sie jeweils auf den Eintrag der Bestellnummer und des Ausgabestandes
   Unter dem Ausgabestand sehen sie die PROFINET Baugruppe CBE30-PN eingeblendet.
   Sobald Sie die CBE30-PN auswählen, verfärbt sich X1400 grün.



4. Ziehen Sie die CBE30-PN auf die entsprechende Schnittstelle der SIMOTION Baugruppe (X1400). Es öffnet sich das Fenster **Eigenschaften - Ethernet Schnittstelle CBE30-PN (R0/S2.6).**
5. Klicken Sie auf **Neu** um ein neues Subnetz anzulegen. Der Dialog **Eigenschaften - Neues Subnetz Industrial Ethernet** wird eingeblendet.
6. Klicken Sie **OK** um die Eingaben zu bestätigen. Es wird ein neues Ethernet Subnetz angelegt, z.B. Ethernet(1).



7. Wählen Sie das Subnetz aus.
8. Vergeben sie die gewünschte IP-Adresse.
9. Übernehmen Sie die Einstellungen mit **OK**.

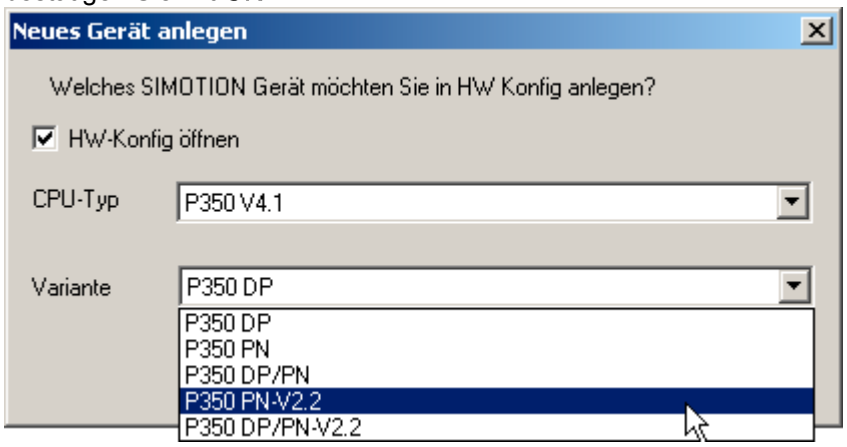
7.3.6 P350 einfügen und projektieren

Voraussetzung

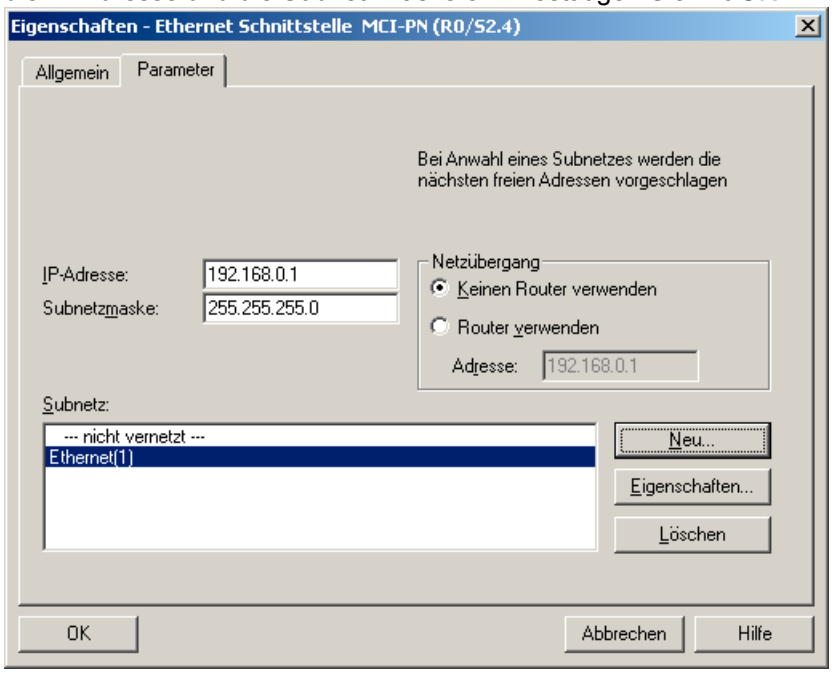
Sie haben bereits ein Projekt angelegt und möchten jetzt eine P350 mit PROFINET einfügen.

Vorgehensweise

1. Klicken Sie auf **Neues Gerät anlegen**, um den Geräte-Auswahldialog zu öffnen.
2. Wählen Sie die verwendete Variante der P350 aus, also P350 PN oder P350 DP/PN) und bestätigen Sie mit **OK**.



3. Der Dialog zum Anlegen eines PROFINET-Subnetzes wird eingeblendet. Geben Sie hier die IP-Adresse und die Subnetzmaske ein. Bestätigen Sie mit **OK**.



4. Wählen Sie im folgenden Dialog die PG/PC-Schnittstelle aus und bestätigen Sie mit **OK**

   HW Konfig wird geöffnet und zeigt die Baugruppe mit dem projektierten PROFINET-Subnetz an.

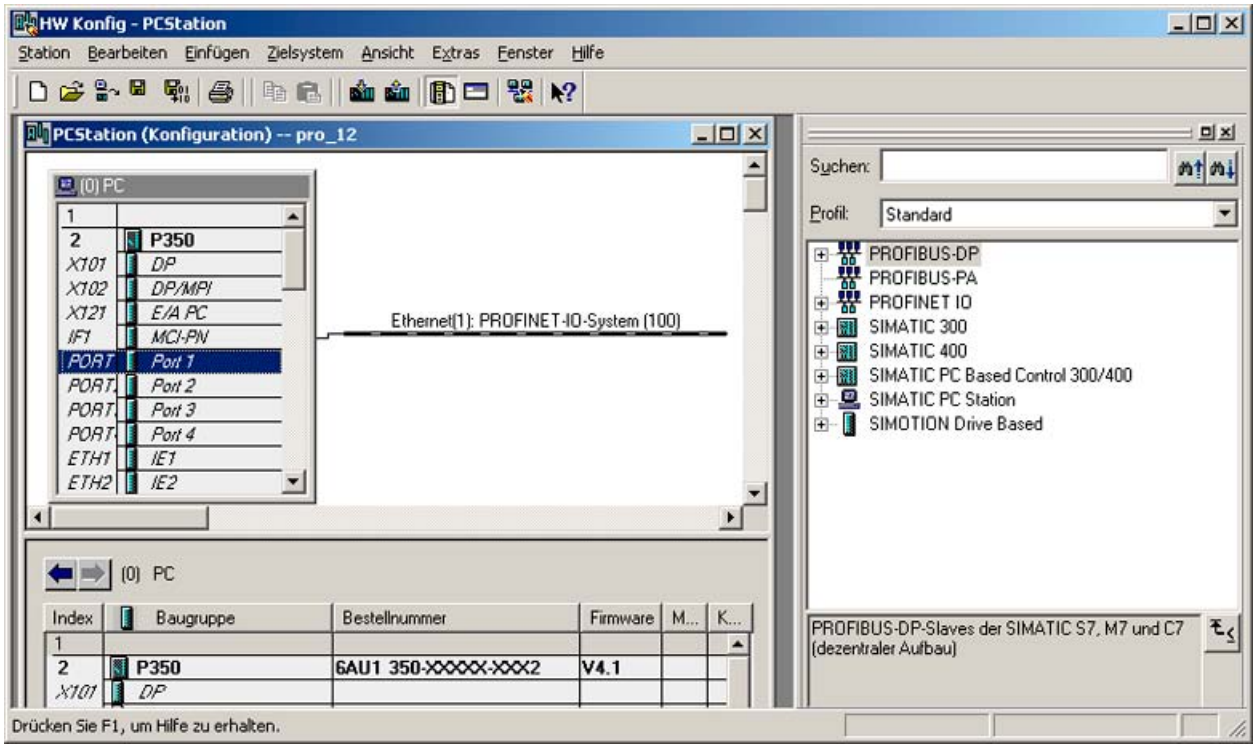


Bild 7-15 HW Konfig mit PROFINET für P350

7.3.7 C240 einfügen und projektieren

Voraussetzung

Sie haben bereits ein Projekt angelegt und möchten jetzt eine C240 mit PROFINET einfügen.

Vorgehensweise

1. Klicken Sie auf **Neues Gerät anlegen**, um den Geräte-Auswahldialog zu öffnen.
2. Wählen Sie die verwendete Variante der C240 PN aus und bestätigen Sie mit **OK**.

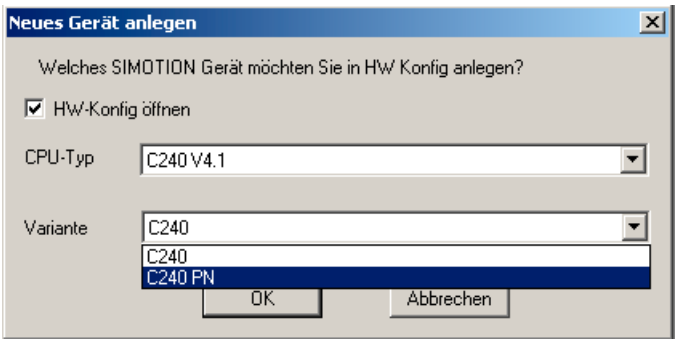


Bild 7-16 Neues Gerät C240 PN anlegen

3. Der Dialog zum Anlegen eines PROFINET-Subnetzes wird eingeblendet. Geben Sie hier die IP-Adresse und die Subnetzmaske ein. Bestätigen Sie mit **OK**.

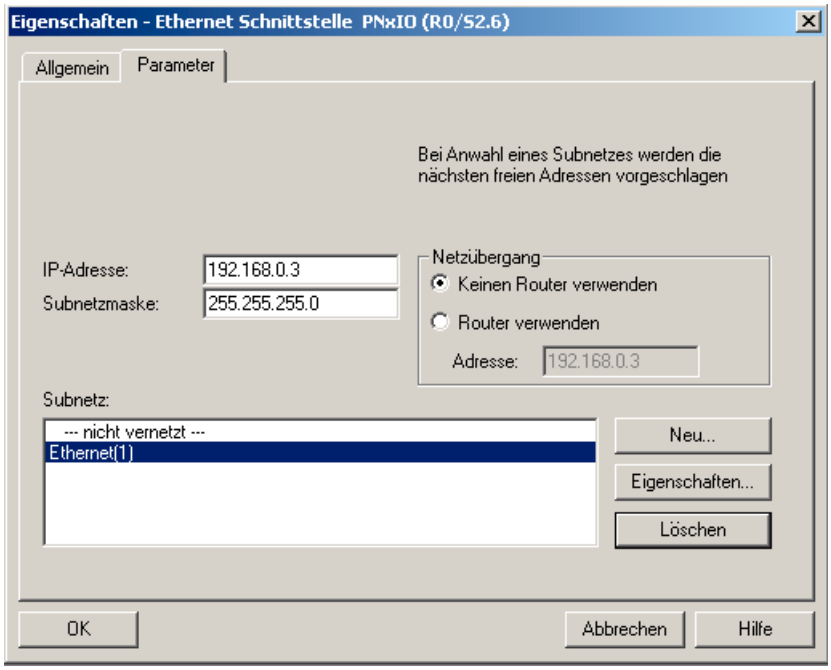


Bild 7-17 Neues Ethernet für C240 PN anlegen

4. Wählen Sie im folgenden Dialog die PG/PC-Schnittstelle aus und bestätigen Sie mit **OK**

   HW-Konfig wird geöffnet und zeigt die Baugruppe mit dem projektierten PROFINET-Subnetz an.

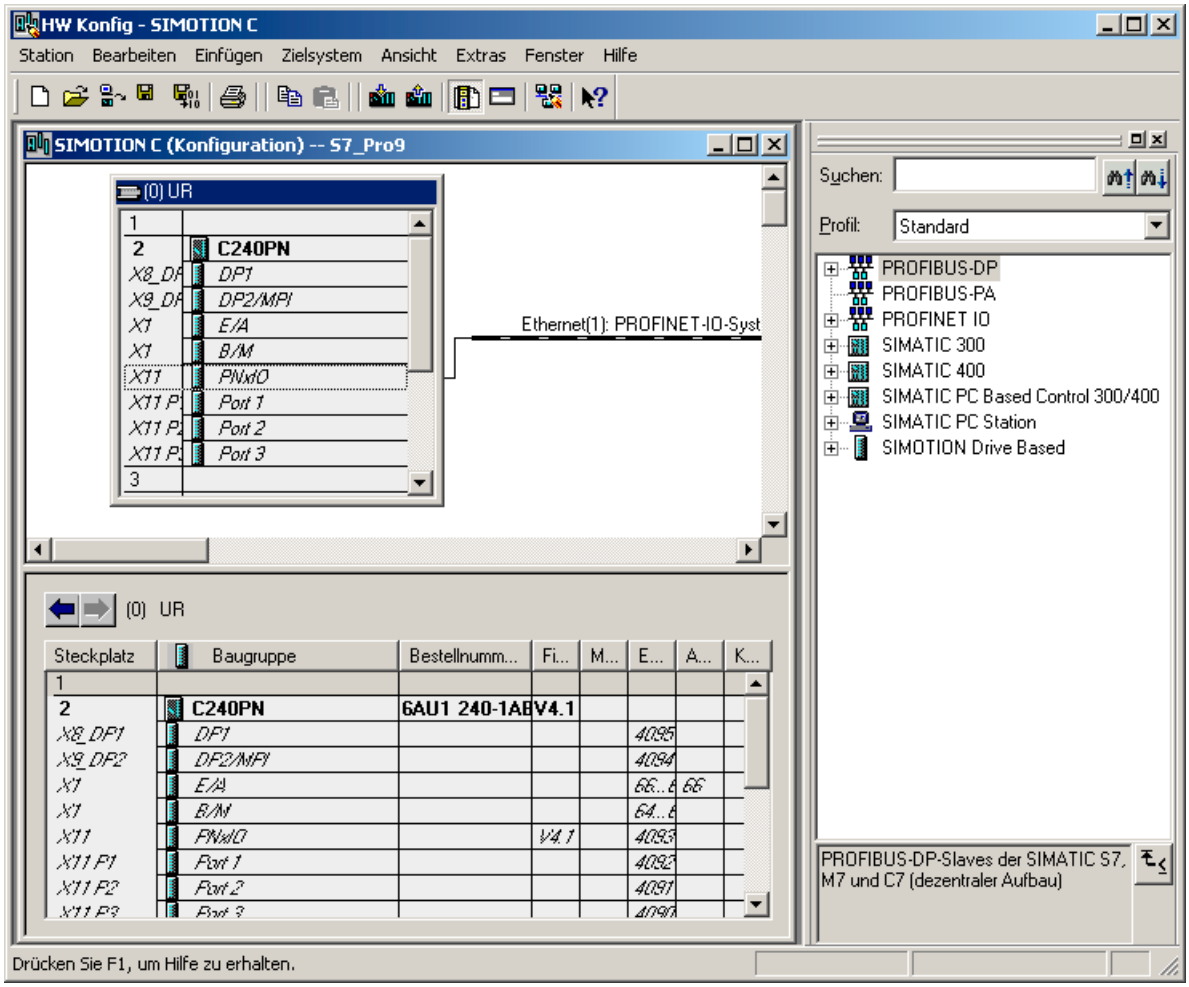


Bild 7-18 HW Konfig mit PROFINET für C240 PN

7.3.8 Sync-Domain anlegen

Eine Sync-Domain ist eine Gruppe von PROFINET-Geräten, die auf einen gemeinsamen Takt synchronisiert sind. Ein Gerät hat die Rolle des Sync-Masters (Taktgeber), alle anderen Geräte haben die Rolle eines Sync-Slaves.

Hinweis

Alle Geräte, die über IRT Daten austauschen, müssen einer Sync-Domain angehören und direkt miteinander verbunden sein, d. h. die Verbindung darf nicht durch nicht IRT-fähige unterbrochen werden, da sonst die Synchronisationsinformationen nicht weitergeleitet werden.

Vorgehensweise

1. Öffnen Sie in HW Konfig die Station mit PROFINET-Geräten, die an der IRT-Kommunikation teilnehmen sollen und markieren Sie z. B. in der Station die PROFINET-Schnittstelle CBE30.
2. Wählen Sie den Menübefehl **Bearbeiten > PROFINET IO > Domain Management**. Ein Dialogregister mit der Liste aller Geräte wird geöffnet. Eine Default Sync-Domain ist angelegt und die Geräte sind bereits zugeordnet.
3. Wählen Sie im oberen Feld die Station aus und doppelklicken Sie im unteren Feld auf das Gerät, das als Sync-Master projektiert werden soll, z. B. CBE30. Der Eigenschaftsdialog des Gerätes wird geöffnet.

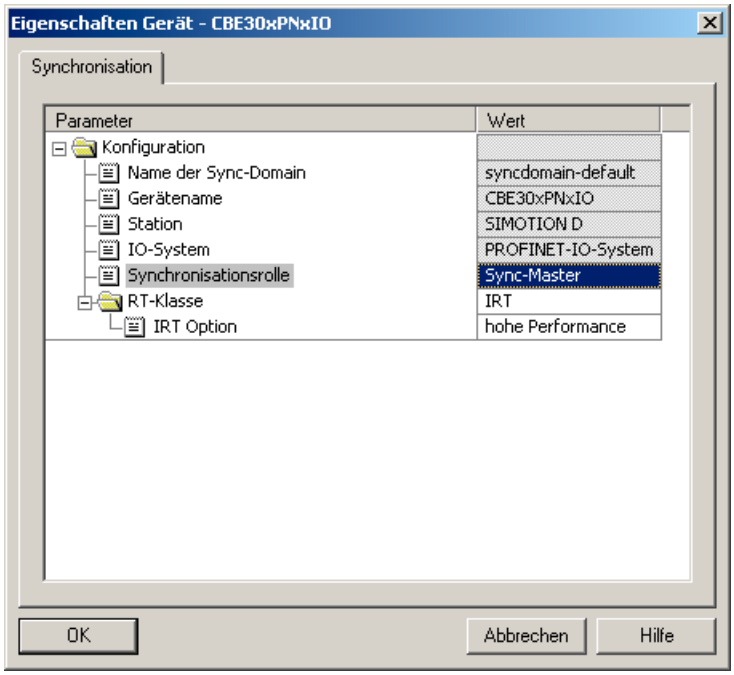


Bild 7-19 Synchronisierung auswählen

4. Stellen Sie die Synchronisationsart auf Sync-Master ein.
5. Quittieren Sie die Einstellungen mit **OK**.
6. Markieren Sie anschließend alle Geräte, die als Sync-Slaves projektiert werden sollen (Strg-Taste gedrückt halten und die Geräte nacheinander markieren).
7. Klicken Sie anschließend auf die Schaltfläche **Eigenschaften** Gerät.
8. Stellen sie im Dialog die Synchronisationsart Sync-Slave ein.
9. Quittieren Sie die Einstellungen mit **OK**.

Hinweis

Alle Geräte, für die **nicht synchronisiert** ausgewählt ist, nehmen nicht an der IRT-Kommunikation, aber automatisch an der RT-Kommunikation teil.

7.3.9 Sendetakt und Aktualisierungszeiten festlegen

PROFINET RT und IRT ist eine zyklische Kommunikation, der Grundtakt ist der Sendetakt. Die Aktualisierungszeit ist ein Vielfaches (2^n) des Sendetakts und in diesem Takt werden die Geräte mit Daten versorgt. Die Aktualisierungszeit kann für jedes Gerät individuell eingestellt werden.

Hinweis

Für PROFINET RT und IRT Hohe Flexibilität ist die Aktualisierungszeit einstellbar, bei IRT Hohe Performance werden die Geräte in jedem Sendetakt mit Daten versorgt (Sendetakt=Aktualisierungszeit).

Für den SINAMICS kann jedoch über den Controller Applikationszyklus definiert werden, dass der SINAMICS nur alle n-Zyklen mit neuen Daten versorgt wird.

Vorgehensweise Sendetakt einstellen

1. Öffnen Sie in HW Konfig den Dialog **Domain Management**.

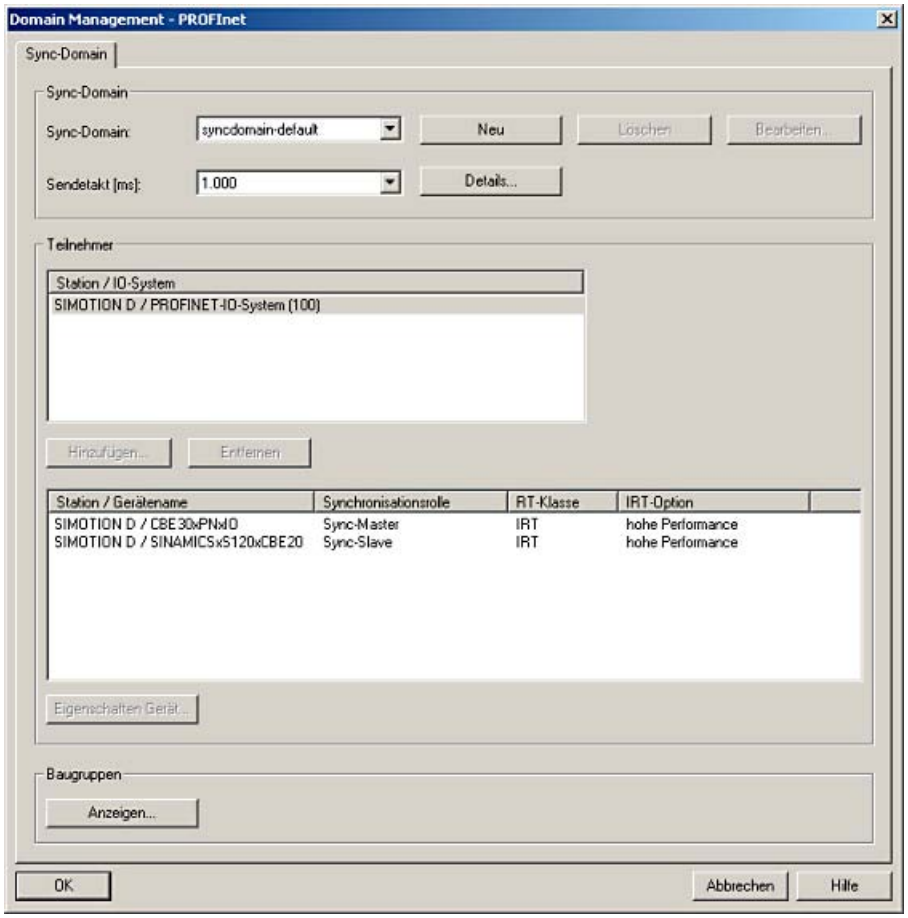


Bild 7-20 Domain Management

2. Wählen Sie einen an den Prozess angepassten Sendetakt. Der Sendetakt ist das kleinstmögliche Sendeintervall. Der Sendetakt ist auf 1 ms vor eingestellt.

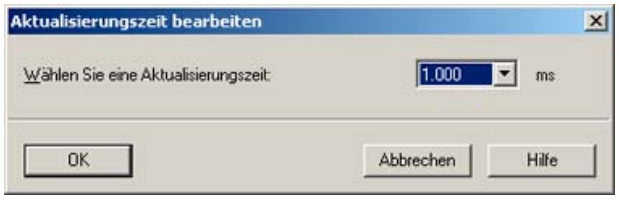
Hinweis

Es sollte nicht so schnell wie möglich sondern nur so schnell wie nötig kommuniziert werden. Dies reduziert den Bandbreitenbedarf und entlastet die Geräte.

Aktualisierungszeiten für PROFINET IO mit PROFINET Geräten mit RT oder IRT Hohe Flexibilität festlegen

Die Aktualisierungszeiten für den IO-Datenaustausch von PROFINET IO mit PROFINET Geräten mit RT bzw. IRT Hohe Flexibilität wird im Dialog **Eigenschaften PROFINET IO-System** eingestellt.

1. Klicken Sie in HW Konfig auf den Strang des PROFINET IO-Systems und wählen Sie im Kontextmenü **Objekteigenschaften**. Der Dialog wird aufgeblendet.
2. Wechseln Sie in das Register **Aktualisierungszeit** und markieren Sie das Gerät in der Übersicht aller IO-Devices.
3. Klicken Sie auf **Bearbeiten**. Im Dialog **Aktualisierungszeit bearbeiten** können Sie die Aktualisierungszeit auswählen.



4. Bestätigen Sie mit **OK**.

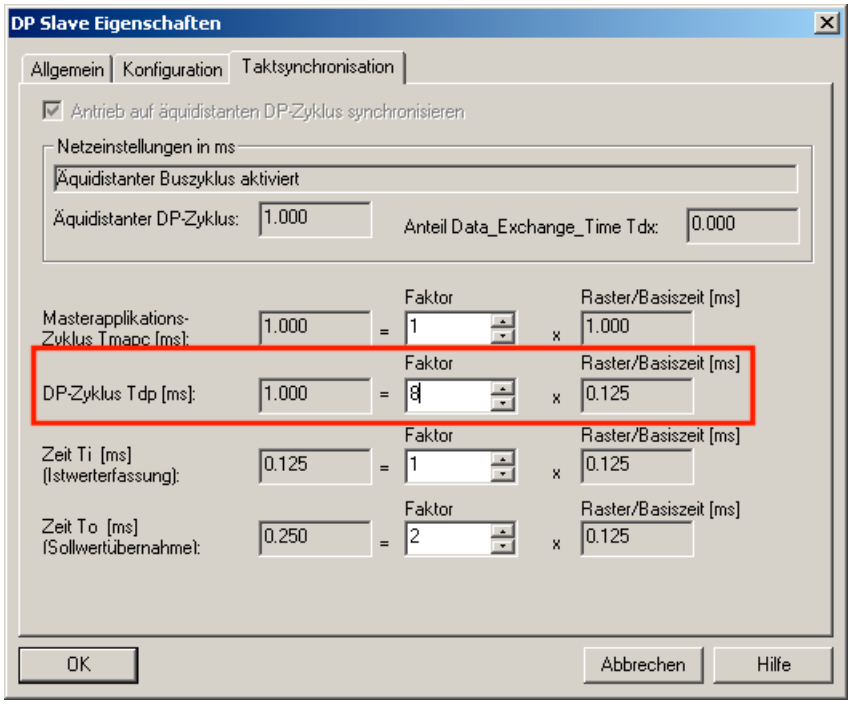
Verhältnis Sendetakt zum DP-Zyklus bei SINAMICS_Integrated (SIMOTION D) mit IRT Hohe Performance

Wird ein SINAMCIS Antrieb an einer SIMOTION D über PROFINET IO IRT Hohe Performance betrieben, muss der DP-Zyklus des integrierten PROFIBUS den Servo Zyklus entsprechen. Per Default ist der DP-Zyklus auf 3 ms eingestellt. Dies entspricht in der Regel nicht dem gewählten Servotakt bei PROFINET Applikationen.

Den DP-Zyklus stellen Sie in den DP Slave Eigenschaften ein:

1. Doppelklicken Sie in HW Konfig auf den SINAMICS_Integrated. Das Fenster **DP Slave Eigenschaften** wird aufgeblendet.
2. Wechseln Sie in das Register **Taktsynchronisation** und aktivieren Sie die Checkbox Antrieb auf äquidistanten DP-Zyklus synchronisieren.

3. Stellen Sie den Faktor beim **DP-Zyklus Tdp** ein. Der DP-Zyklus muss gleich dem Servo Takt sein. Ist der Servo Takt z. B. 1 ms, müssen Sie als Faktor 8 eintragen (8 x [Basiszeit 0.125 ms] = 1 ms).



4. Bestätigen Sie mit **OK**.

Wenn PROFINET taktsynchron betrieben wird, muss der Servo-Takt immer dem PROFIBUS-Takt entsprechen. Servo-Takt und PROFIBUS-Takt können zum PROFINET-Takt untersetzt werden.

Beispiel:

PROFINET-Sendetakt = 0,5 ms

PROFIBUS-Takt = Servo-Takt = 1 ms

Der PROFIBUS-Takt kann zum PROFINET-Takt im Verhältnis 1:1 bis 1:16 betrieben werden.

7.3.10 Topologie projektieren

7.3.10.1 Topologie

Einleitung

Voraussetzung für IRT Hohe Performance ist die Topologieprojektierung und die Angabe welches Gerät über welchen Port mit welchen anderen Geräten verbunden ist.

Hinweis

Bei IRT Hohe Flexibilität bzw. RT kann die Topologie optional projektiert werden z. B. für die topologiebasierte Taufe oder für Diagnosezwecke.

Zur Festlegung der Eigenschaften von Leitungen zwischen den Ports der Switche haben Sie zwei Möglichkeiten:

Über den Topologie-Editor (Seite 188)

Über die Objekteigenschaften

7.3.10.2 Topologie-Editor (Graphische Ansicht)

Vorgehensweise

Mit dem Topologieeditor haben Sie eine Übersicht über sämtliche Ports im Projekt und können sie zentral verschalten.

Den Topologieeditor starten Sie mit dem Menübefehl Bearbeiten > PROFINET IO > Topologie in HW Konfig oder NetPro (PROFINET-Gerät muss ausgewählt sein).

Im Topologieeditor haben Sie zwei Möglichkeiten die Topologie graphisch (ab STEP7 V5.4 SP2) oder tabellarisch anzuzeigen. Zum Verschalten ist die graphische Ansicht besser geeignet.

Beschreibung

Im Topologie Editor können Sie:

- Ports verschalten
- Eigenschaften der Verschaltung anpassen
- Passive Komponenten einfügen
- Im Online-Modus einen Vergleich Offline/Online anzeigen lassen

Vorgehensweise

1. Doppelklicken Sie im SCOUT auf die SIMOTION Baugruppe um HW Konfig aufzurufen.
2. Wählen Sie die PROFINET-Baugruppe aus, z. B. eine CBE30-PN.
3. Führen Sie **Bearbeiten > PROFINET IO > Topologie** aus. Der Toplogie-Editor wird eingeblendet.
4. Klicken Sie auf **Grafische Ansicht**, um die Registerkarte in den Vordergrund zu schieben.

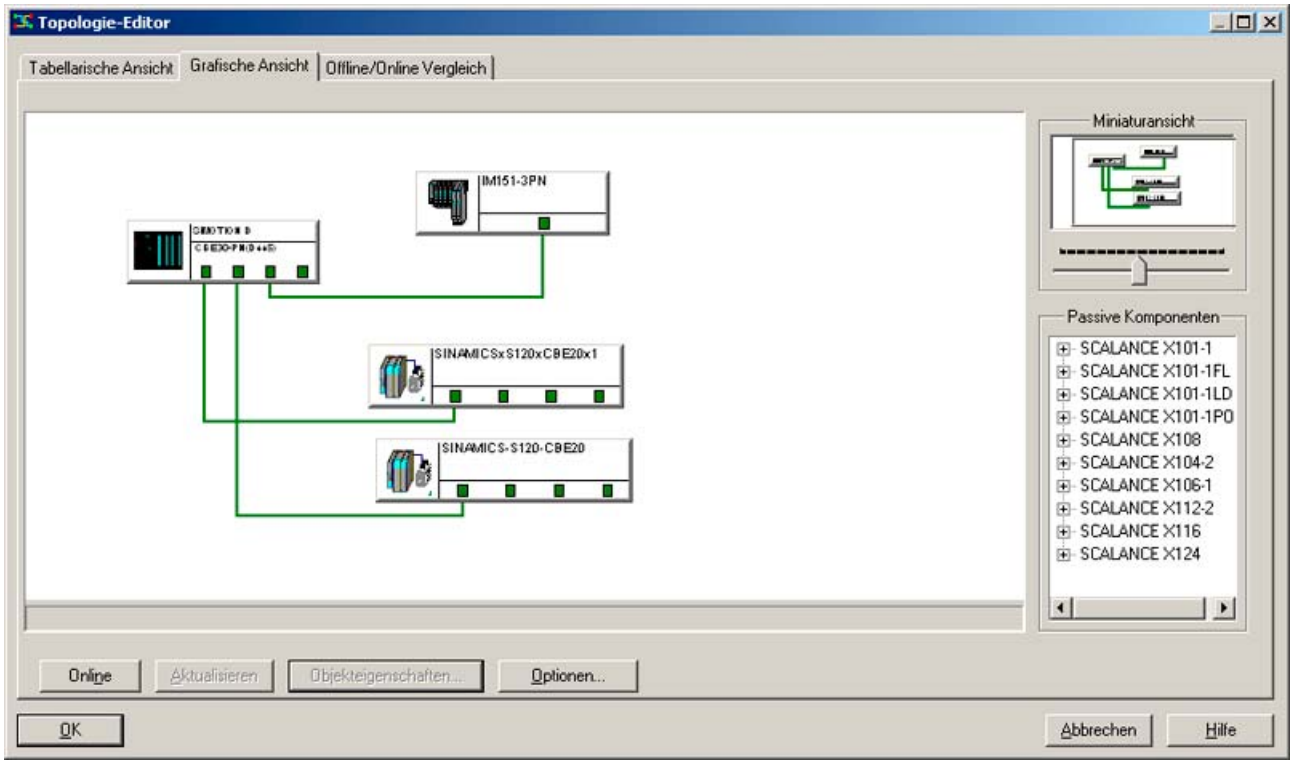
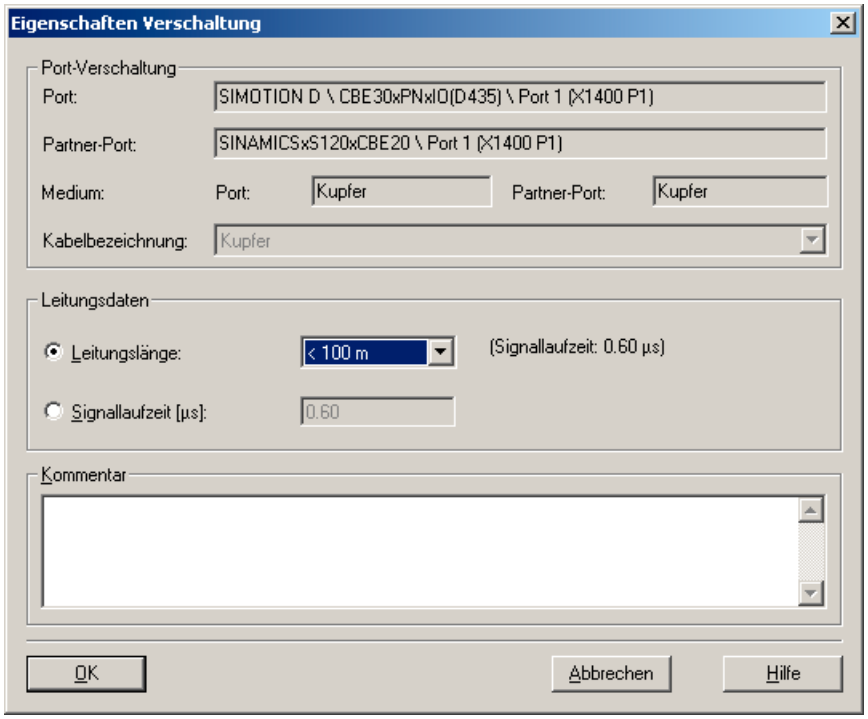


Bild 7-21 Topologie-Editor (grafische Ansicht)

5. Verbindungen zwischen Ports erstellen Sie, in dem Sie bei gedrückter linker Maustaste eine Verbindung zwischen den beiden Ports ziehen. Das Fenster **Eigenschaften Verschaltung** wird geöffnet.

6. Es wird Ihnen die Port-Verschaltung angezeigt. Sie haben die Möglichkeit die Leitungsdaten zu projektieren. Defaultmäßig wird eine Leitungslänge von <100m eingestellt, die empfohlen wird. Alternativ haben Sie die Möglichkeit die Signallaufzeit zu projektieren.



7. Bestätigen Sie mit **OK**.

Offline/Online-Vergleich

Wenn Sie in den Online-Modus wechseln wird die Topologie im Editor mit der realen Topologie verglichen. Nicht bekannte Komponenten werden mit Fragezeichen angezeigt, Verbindungen und Komponenten im RUN werden grün markiert.

7.3.10.3 Ports über den Topologie-Editor(tabellarische Ansicht) verschalten

Vorgehensweise

In der Tabellarischen Übersicht des Topologieeditors können Sie Ports verschalten.

Den Topologieeditor starten Sie mit dem Menübefehl **Bearbeiten > PROFINET IO > Topologie** in HW Konfig oder NetPro (PROFINET-Gerät muss ausgewählt sein).

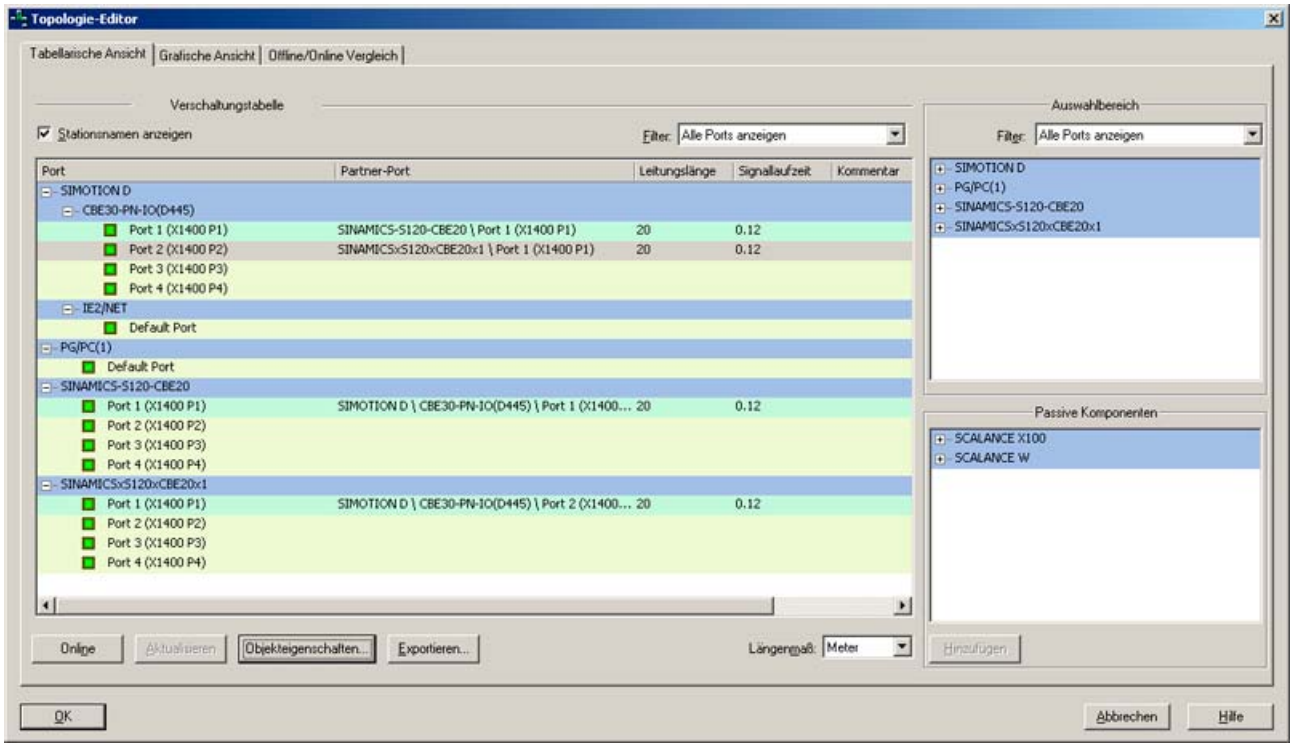


Bild 7-22 Topologie Editor

In der Verschaltungstabelle auf der linken Seite werden alle projektierten PROFINET IO Geräte mit ihren Ports aufgelistet. Über den Filter-Dialog können Sie wählen, ob alle Ports, nur die noch nicht verschalteten Ports oder nur die bereits verschalteten Ports angezeigt werden sollen.

Verschaltung von Ports in der tabellarischen Übersicht

1. Zur Verschaltung von Ports unterschiedlicher Geräte wählen Sie im rechten Auswahlbereich den Port eines Gerätes aus, den Sie verschalten möchten.
2. Ziehen Sie diesen Port auf den gewünschten Port eines Gerätes in der Verschaltungstabelle. Darauf öffnet sich der Dialog Eigenschaften Verschaltungen.
3. Projektieren Sie die Leitungsdaten. Defaultmäßig sollte eine Leitungslänge von <100m eingestellt werden.
4. Bestätigen Sie Ihre Eingabe mit **OK**.

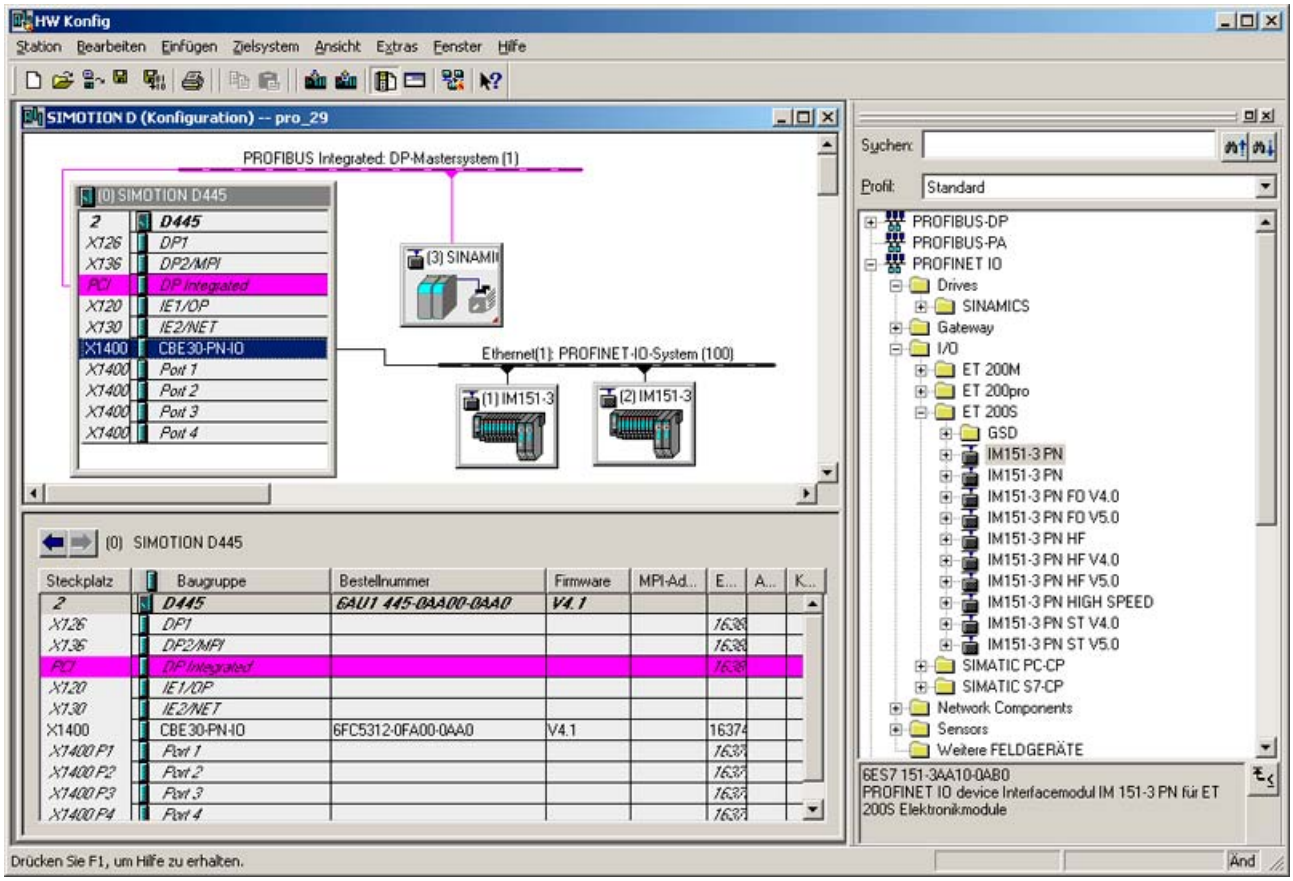
7.3.11 IO-Device anlegen

Voraussetzung

Sie haben bereits ein PROFINET IO-System angelegt und eine PROFINET IO-Baugruppe, z.B. SIMOTION D445 mit CBE30-PN projektiert (siehe SIMOTION CPU D4x5 einfügen und projektieren (Seite 172)).

Vorgehensweise bei PROFINET IO-Devices über den Hardware Katalog

1. Doppelklicken Sie in SIMOTION SCOUT auf die entsprechende Baugruppe um HW Konfig zu öffnen.
2. Wählen Sie im Baugruppenkatalog unter PROFINET IO die Baugruppe aus, die Sie an das PROFINET IO-System anschließen möchten.
3. Ziehen Sie die Baugruppe auf den Strang des PROFINET IO Systems. Das IO-Device wird eingefügt.



4. Speichern und übersetzen Sie die Einstellungen in HW Konfig.

Vorgehensweise bei PROFINET IO-Devices von Fremdherstellern

1. Doppelklicken Sie auf die entsprechende Baugruppe um HW Konfig zu öffnen.
2. Wählen Sie den Menübefehl **Extras > GSD-Dateien** installieren.
3. Wählen Sie im Dialog GSD-Dateien installieren die zu installierende GSD-Datei aus.
4. Klicken Sie auf den Button **Installieren**.
5. Schließen Sie den Dialog durch Klicken des Buttons **Schließen**.
6. Wählen Sie im Baugruppenkatalog unter PROFINET IO die Baugruppe aus, die Sie an das PROFINET IO-System anschließen möchten.
7. Ziehen Sie die Baugruppe auf den Strang des PROFINET IO System. Das IO-Device wird eingefügt.
8. Speichern und übersetzen Sie die Einstellungen in HW Konfig.

7.3.12 SINAMICS S120 einfügen und projektieren

Voraussetzung

Sie haben in Ihrem Projekt ein SIMOTION Gerät und eine CBE30 eingefügt und es ist bereits ein PROFINET IO Subnetz angelegt.

Vorgehensweise

1. Wählen Sie im Hardware-Katalog von HW Konfig den Eintrag **PROFINET IO > Drives > SINAMICS** und dort die Baugruppe, z. B. **SINAMICS S120 CU320 CBE20**.
2. Klicken Sie auf den Eintrag und ziehen den Antrieb z. B. **V2.5 PN-V2.2** auf das PROFINET IO Subnetz. Es öffnet sich das Fenster **Eigenschaften - Ethernet Schnittstelle SINAMICS-S120-CBE20**.
   Dort ist bereits eine IP-Adresse vorgeschlagen und das Subnetz ausgewählt.
3. Klicken Sie **OK**, um die Einstellung zu übernehmen.
   Es wird der Dialog **Eigenschaften SINAMICS** eingeblendet.

4. Wählen Sie die Geräteversion (Firmware-Version) aus.

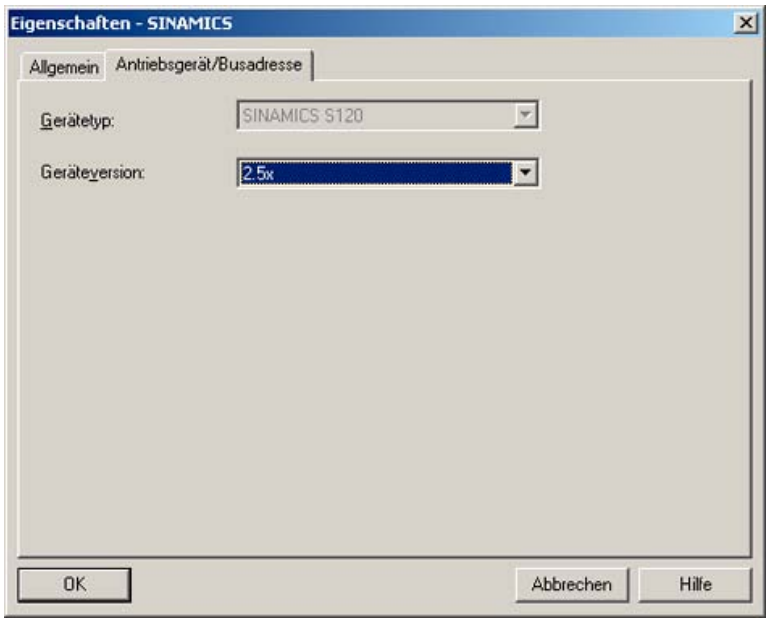


Bild 7-23 Eigenschaften SINAMICS

5. Klicken Sie **OK** um die Eingaben zu bestätigen.
6. Wählen Sie in HW Konfig **Station > Speichern und alles übersetzen**.

Telegramm einstellen

Falls Sie taktsynchrone Kommunikation nutzen möchten, müssen Sie ein Telegramm projektieren, mit dem dies möglich ist (z. B. Telegramm 105), z. B. damit der Antrieb auf PROFINET aufsynchronisiert werden kann. Nur bei taktsynchronem Antrieb sind die Eingabefelder Ti oder To im Dialog **Eigenschaften SINAMICS CBE20 PN-IO** aktiv.

Das Telegramm wird standardmäßig im SCOUT bei der Konfiguration des Antriebsgerätes eingestellt.

1. Klicken Sie im SCOUT im Projektnavigator und dem Antriebsgerät auf **Antriebsgerät konfigurieren**, falls Sie noch keine Einspeisung und Antrieb projektiert haben.
2. Durchlaufen Sie den Assistenten zum Konfigurieren des Antriebsgerätes. Für PROFINET steht eine Onboard-Schnittstelle zur Verfügung, der zwei Interfaces PZD IF1 und PZD IF2 zugeordnet sind. Bei der Konfiguration können Sie diese festlegen. Es wird empfohlen die Einstellung **Automatisch** zu belassen. Bei den Telegrammen sollten Sie SIEMENS Telegramme XXX wählen z. B. beim Antrieb Telegramm 105.
3. Nachdem Sie den Assistenten abgeschlossen haben, klicken Sie im Projektnavigator unterhalb des Antriebsgeräts auf **Kommunikation > Telegrammkonfiguration**. Im Register **IF1: PROFIdrive PZD-Telegramme** sind die Telegramme aufgelistet.
4. Klicken Sie auf **Übertrage nach HW Konfig**, um die Adressen abzugleichen. Nach dem Abgleich wird hinter dem Adressbereichen ein Haken angezeigt.
5. Wählen Sie im Menü **Projekt > Speichern**.

Telegramm in HW Konfig

Alternativ können Sie das Telegramm auch in HW Konfig zuweisen. Dazu müssen das Antriebsgerät und der Antrieb im SCOUT schon projektiert sein. Dann können Sie in HW Konfig wie folgt vorgehen:

1. Wählen Sie den eingefügten SINAMICS-Antrieb aus und doppelklicken Sie in der unteren Tabelle auf den Eintrag **SIEMENS / Standard Telegramm xx**.
   Der Dialog **Eigenschaften SIEMENS / Standard Telegramm xx** wird aufgerufen.
2. Wählen Sie das entsprechende Telegramm aus. Nach dem Speichern kann das Telegramm dann auch im SCOUT im Projektnavigator unter **<"Antriebsgerät_xx"> - Kommunikation > Telegrammkonfiguration** ausgewählt werden. Ein Abgleich mit HW Konfig ist möglich.

Einstellungen an der SINAMICS PROFINET Schnittstelle

Damit der eingefügte SINAMICS S120 Antrieb taktsynchron im PROFINET läuft, müssen noch Einstellungen am SINAMICS durchgeführt werden.

1. Wählen Sie den SINAMICS Antrieb am PROFINET IO System aus und doppelklicken Sie in der unteren Tabelle auf den Eintrag des PROFINET Interfaces, z.B. CBE20-PN-IO.
   Der Dialog **Eigenschaften CBE20-PN--IO** wird eingeblendet.
2. Wählen Sie das Register **Applikation** aus.

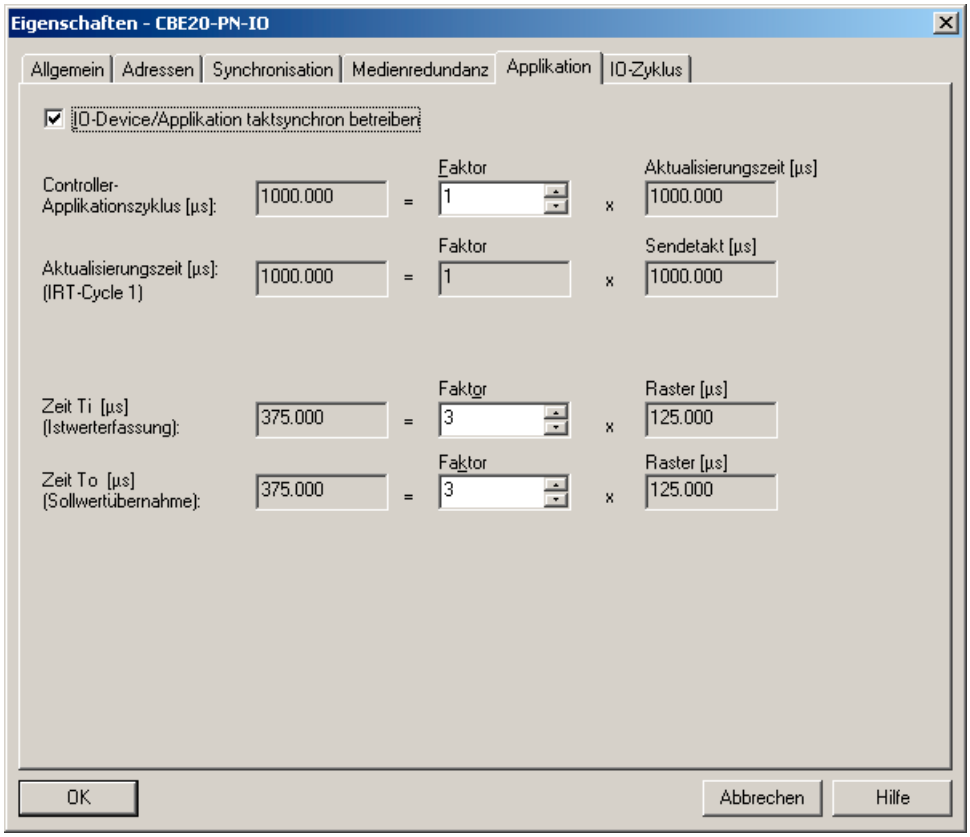


Bild 7-24 Eigenschaften SINAMICS CBE20 PN-IO

3. Klicken Sie den Eintrag **IO-Device/Applikation taktsynchron betreiben** an.
   Der Antrieb nimmt jetzt an der taktsynchronen Kommunikation teil.

4. Geben Sie ggf. unter **Controller-Applikationszyklus** einen Wert > 1 ein, um Taktuntersetzung zu projektieren. Dies ist notwendig, wenn der Servo langsamer als der IRT-Takt ist. Der Servo muss jedoch im IRT-Takt fertig berechnet sein.
5. Wechseln Sie ggf in das Register **Synchronisation**, um die **Synchronisationsart**, hier **Sync-Slave** auszuwählen. Das kann auch unter **Domain Management** eingestellt werden.
6. Bestätigen Sie die Eingaben mit **OK**.

Hinweis

Ein taktsynchroner Betrieb ist nur möglich, wenn PROFINET IRT Hohe Performance projektiert wurde.

Als weitere Arbeitsschritte müssen Sie den Antrieb in die Sync-Domain einbinden (siehe Sync-Domain anlegen), die IP-Adresse in den Antrieb laden, siehe Gerätenamen und IP-Adressen für IO-Devices vergeben (Seite 195)und die Ports verschalten (siehe Ports über den Topologie-Editor(tabellarische Ansicht) verschalten (Seite 188)).

Siehe auch

Sync-Domain anlegen (Seite 179)

7.3.13 IP-Adresse und Kommunikationsname

Einleitung

Ein Gerät bzw. Controller besitzt eine feste MAC-, eine projektierbare IP-Adresse und einen Kommunikationsnamen. Die IP-Adresse, Subnetzmaske und den Kommunikationsnamen legen Sie im Engineering unter den Eigenschaften der Ethernet Schnittstelle fest. Damit besitzen die Geräte im Projekt eine eindeutige Zuordnung.

Den Kommunikationsnamen (Gerätenamen) finden Sie Dialog Eigenschaften - Schnittstelle, wenn Sie auf das Gerät bzw. den Controller Doppelklicken. Klicken Sie im Dialog auf Eigenschaften, um die IP-Adresse festzulegen. Siehe auch dazu CBE30-PROFINET-Board einfügen und projektieren (Seite 173) .

Richtlinien für Kommunikationsnamen

Beim Device ist der Kommunikationsname gleich dem Gerätename z. B. *Drive1* bei einem Antrieb. Bei Controllern ist der Kommunikationsname der Name des PROFINET-Interfaces z. B. *CBE30xPNxIO* bei einer CBE30 an SIMOTION D.

Kommunikationsnamen unterliegen bestimmten Syntaxregeln d. h. Sie müssen grundsätzlich DHCP-konform sein und für SIMOTION und SIMATIC existieren noch weitere Randbedingungen.

- Buchstaben und Ziffer sind zulässig a-z und 0-9
- keine Sonderzeichen ! " § $ % & / ( ) = ? * ' _ : ; > < , # + | ~ \ }] [{

- Der Name kann sich aus mehreren Teilen zusammen setzen
  - Label.Label.Label.Label
  - Der Punkt dient als Trennzeichen
  - Das Label muss mit einem Buchstaben beginnen bzw. enden
  - Die maximale Länge des Label sind 63 Zeichen
  - Labels darf nicht mit "xn-" beginnen
- Die maximale Gesamtlänge des Namens beträgt 240 Zeichen
- Reservierte Namen "port-xyz" oder "port-xyz-abcde"
- Es wird nicht zwischen Groß- und Kleinschreibung unterschieden. Da vom Engineering-System alle Namen mit Kleinbuchstaben angezeigt werden, sollten Sie auch nur diese verwenden.
- Im Zusammenhang mit SCOUT darf das Minus "-" nicht verwendet werden
- Nicht zulässige Zeichen werden vom Engineering durch ein x ersetzt.

Taufe von Controller und Devices im Online-Modus

Der Controller und die Devices besitzen im Auslieferungszustand noch keinen Kommunikationsnamen und keine IP-Adresse, diese müssen erst zugewiesen werden. Bei der Adressvergabe d. h. IP-Adresse und Kommunikationsname zuweisen, auch Taufe genannt, wird zwischen Controllern und Devices unterschieden. Devices und Controller können Sie auf unterschiedliche Arten taufen.

Controller Taufe

- Download der Applikation
- Engineering Software
  - HW Konfig, NetPro, SCOUT
  - Primary Setup Tool (PST)
- Durch die Applikation (Systemfunktion _setNameofStation für SIMOTION)

Hinweis

Bei der Taufe mit der Engineering Software sollten Sie sich gerade bei größeren Anlagen direkt mit dem Gerät verbinden, da dadurch das zu taufende Gerät eindeutig identifiziert wird. Alternativ steht das Feature Blinken zur Verfügung, in dem das Gerät über eine blinkende LED identifiziert werden kann.

Device Taufe

- Engineering Software
  - HW Konfig, NetPro
  - SCOUT, Starter
  - Primary Setup Tool (PST)
- Vorher die MMC oder CF-Karte beschreiben und dann stecken

Topologiebasierte Taufe für Device

Devices können auch ohne MMC- bzw. CF-Karte getauft werden. Dies wird als topologiebasierte Taufe bezeichnet. Dies wird nur ab bestimmten Softwareständen unterstützt.

- SIMATIC S7-300 FW ≥ V 2.7
- SIMATIC S7-400 FW ≥ V 5.2
- SIMOTION FW ≥ V4.1.2 mit PN V2.2
- SINAMICS FW ≥ V2.5.1.10 mit PN V2.2
- ET200S FW ≥ 6.0
- ET200S HS FW ≥ 2.0

7.3.14 Gerätenamen und IP-Adressen für IO-Devices vergeben

Einleitung

Um auf ein Gerät (Controller oder Device) mit dem Engineering online gehen zu können, muss dem Gerät im Engineering eine IP-Adresse zugewiesen werden. Weiterhin muss den Geräten ein im Netzwerk eindeutiger Kommunikationsname zugewiesen werden. Dieser wird benötigt damit der Controller seine ihm zugeordneten Devices identifizieren kann.

Die IP-Adresse können Sie im Dialog **Eigenschaften - Ethernet Schnittstelle ….** festlegen (kann mit Doppelklick auf das Gerät geöffnet werden). Außerdem wird standardmäßig ein Gerätename eingetragen, den Sie ändern können. Als Standardeinstellung ist die Einstellung **IP-Adresse durch Controller zuweisen** aktiv. D.h. im Hochlauf identifiziert der Controller die ihm zugeordneten Devices über den Kommunikationsnamen und weist ihnen dann die im Engineering festgelegte IP-Adresse zu. Es wird empfohlen diese Funktion nicht zu deaktivieren.

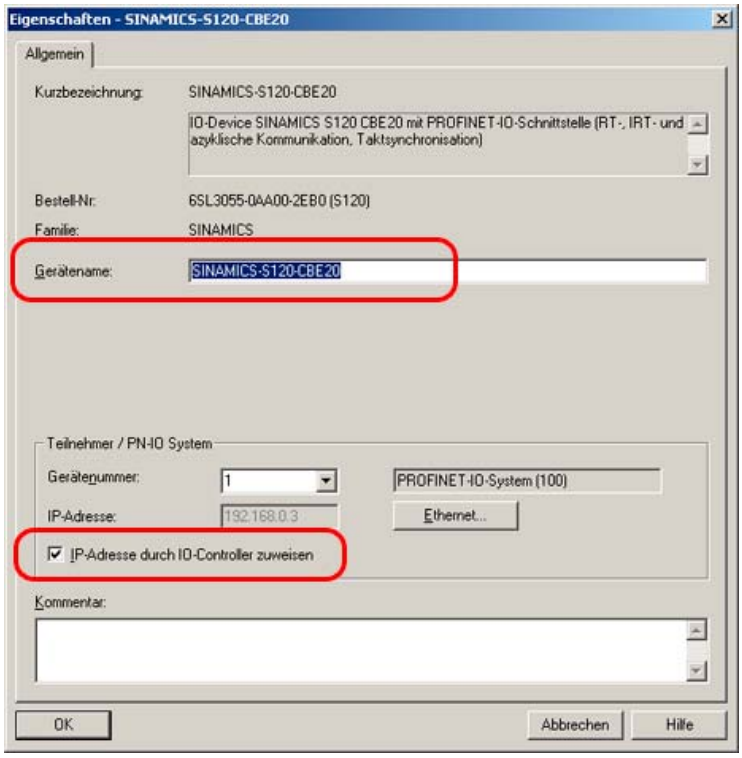


Bild 7-25 Eigenschaften SINAMICS S120

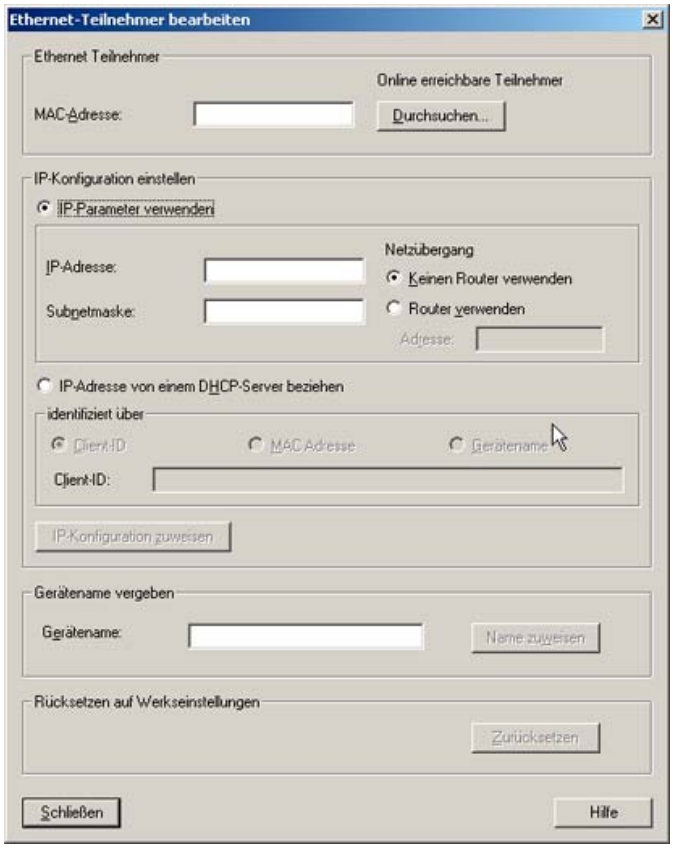
Taufe eines IO-Devices

Hinweis

Wenn Sie das PG/PC direkt an der PROFINET-Schnittstelle des Gerätes anschließen, können Sie ein Patch oder Crossover-Kabel verwenden.

Bei der Inbetriebnahme wird empfohlen sich direkt mit dem zu taufenden Gerät mit dem PG/PC zu verbinden

1. Wählen Sie in HW Konfig oder NetPro den Menüpunkt **Zielsystem - Ethernet - Ethernet-Teilnehmer bearbeiten**. Es öffnet sich der Dialog **Ethernet-Teilnehmer bearbeiten**.



2. Klicken Sie auf den Button **Durchsuchen**.
3. Es öffnet sich der Dialog **Netz durchsuchen**. Die angeschlossenen Teilnehmer werden angezeigt.

4. Klicken Sie das zu taufende Gerät an und bestätigen Sie mit **OK**.

5. Geben Sie die IP-Adresse und Subnet Maske, die Sie im Dialog **Eigenschaften - Ethernet Schnittstelle …** festgelegt haben, an.
6. Die Default Einstellung (**Keinen Router verwenden**) für den Netzübergang bleibt unverändert.
7. Klicken Sie den Button **IP-Konfiguration zuweisen**. Die IP-Adresse wird dann online dem Gerät zugewiesen
8. Geben Sie den Gerätenamen ein, den Sie in HW Konfig festgelegt haben, siehe Bild **Eigenschaften SINAMICS S120**.
9. Klicken Sie den Button **Name** zuweisen. Es wird der Gerätename dem Gerät zugewiesen.

Alternativ Knotentaufe in SIMOTION SCOUT durchführen

Sie können die Knotentaufe auch im SCOUT durchführen.

- Führen Sie im SCOUT **Erreichbare Teilnehmer** aus und klicken Sie im eingeblendeten Dialog mit der rechten Maustaste das Gerät an, das Sie bearbeiten möchten.
- Führen Sie **Ethernet Teilnehmer bearbeiten** aus. Der entsprechende Dialog wird eingeblendet.

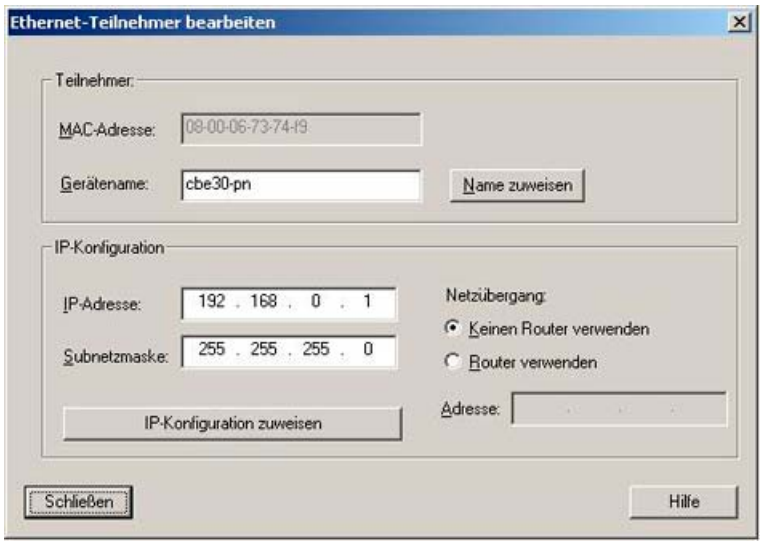


Bild 7-26 Ethernet Teilnehmer bearbeiten

- Geben Sie einen Gerätenamen, eine Subnetzmaske und eine IP-Adresse ein.
- Bestätigen Sie die Eingaben.

Gerätename und IP-Adresse werden in das Gerät übertragen und dort gespeichert.

7.4 Direkten Datenaustausch zwischen IO Controllern projektieren

7.4.1 Einleitung

Zwischen zwei oder mehreren SIMOTION Controllern können E/A-Datenbereiche zyklisch über IRT Hohe Performance ausgetauscht werden. Dies wird auch als Controller Controller Querverkehr bezeichnet. Der Controller Controller Querverkehr ist nur über PROFINET IO mit IRT Hohe Performance zwischen SIMOTION Controllern möglich.

Für den Datenaustausch müssen sich die Geräte in einer gemeinsamen Sync-Domain befinden und entsprechend als Sync-Master und Sync-Slave konfiguriert sein.

Hinweis

Für SIMATIC CPUs steht diese Funktion nicht zur Verfügung.

Es gibt zwei Arten von Querverkehr, den vom System automatisch angelegten Querverkehr z. B. verteilter Gleichlauf und den applikativen, vom Anwender in seiner Applikation nutzbaren Querverkehr. Diesen Querverkehr können Sie projektieren.

Hinweis

Der vom System automatisch projektierte Querverkehr darf nicht durch den Anwender in den Engineering Tools verändert werden (z.B. Veränderung der Adressbereiche). Dies führt zu Fehlerzuständen!

Empfehlung

Wir empfehlen, zunächst die Sendebereiche für alle PROFINET-Geräte zu projektieren und anschließend die Empfangsbereiche. Bei dieser Vorgehensweise können Sie bei der Definition der Empfangsbereiche die zuvor definierten Sendebereiche zuweisen. Damit werden Fehleingaben vermieden.

Datenmenge

Es können ca. 3 KByte übertragen werden. Für jede projektierte Gleichlaufbeziehung werden 24 Byte benötigt. D. h. wurden zu einer Leitachse 5 Folgeachsen definiert, benötigt das System 5 * 24 Byte. Die verbleibende Datenmenge steht für den applikativen Querverkehr zur Verfügung.

Hinweis

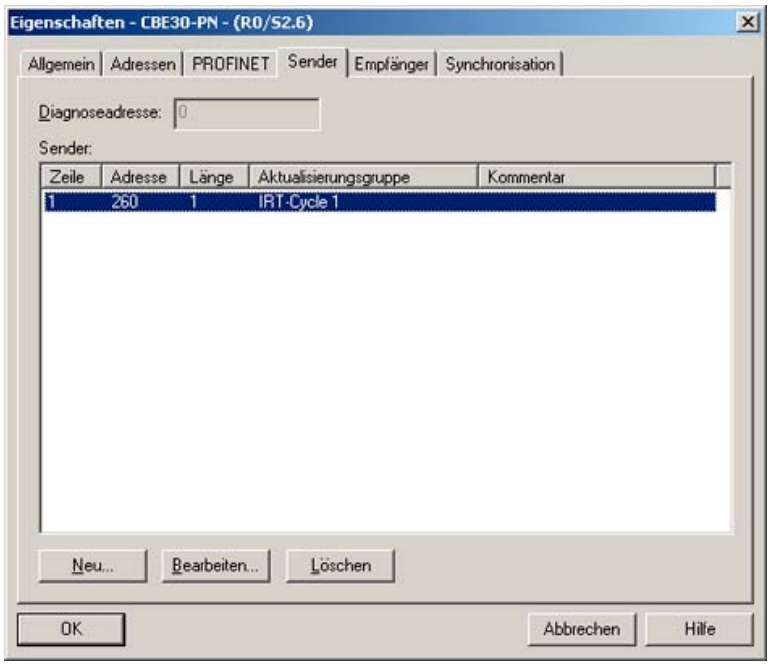
In den SIMOTION Utilities & Applications ist ein FAQ zum Thema PROFINET Projektierung beschrieben. Die SIMOTION Utilities & Applications sind im Lieferumfang von SIMOTION SCOUT enthalten.

In diesem FAQ werden die Themen Verteilter Getriebegleichlauf und Controller / Controller Querverkehr behandelt.

7.4.2 Sender projektieren

Vorgehensweise

1. Öffnen Sie den Eigenschaftsdialog des PROFINET Interfaces (Doppelklick auf die entsprechende Zeile in der Konfigurationstabelle von HW Konfig).
2. Wählen Sie das Register **Sender**.

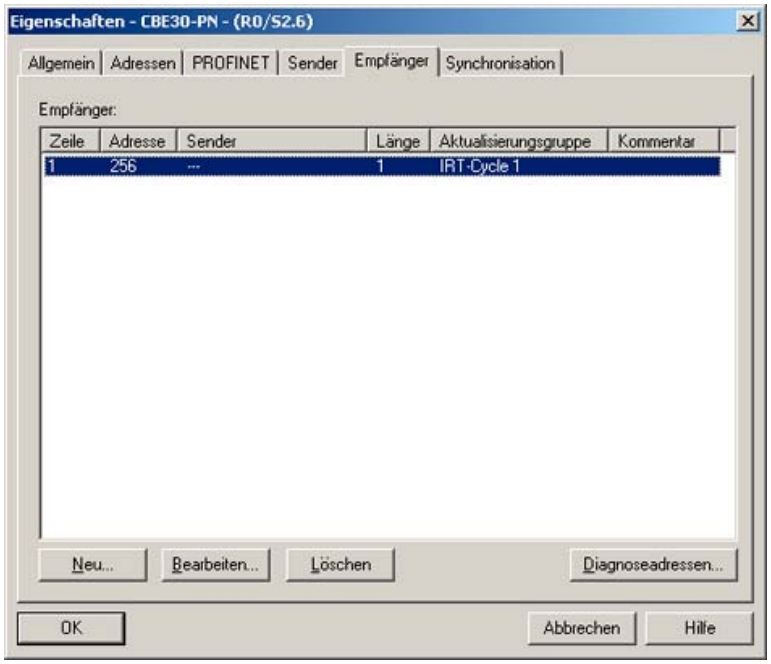


3. Klicken Sie auf die Schaltfläche **Neu**.
4. Geben Sie im Eigenschaftsdialog des Senders Anfangsadresse aus dem I/O-Bereich und Länge des Adressbereichs ein, über den gesendet werden soll. Kommentieren Sie den Datenbereich, um später die über diesen Bereich gesendeten Daten identifizieren zu können. Die maximale Größe einer Variablen ist auf 254 Byte begrenzt.
5. Quittieren Sie die Einstellungen mit **OK**.
6. Wiederholen Sie die Schritte 3 bis 5 für weitere Sendebereiche.
7. Ändern Sie, wenn gewünscht, die voreingestellte Diagnoseadresse für die Sendebereiche.
8. Bestätigen Sie Ihre Eingabe mit **OK**.

Für die Kommunikationsbeziehung, in der ein PROFINET Interface als Sender für direkten Datenaustausch auftritt, ist genau eine Diagnoseadresse zu vergeben.

7.4.3 Empfänger projektieren

Vorgehensweise



1. Öffnen Sie den Eigenschaftsdialog des PROFINET Interfaces (Doppelklick auf die entsprechende Zeile in der Konfigurationstabelle von HW Konfig).
2. Wählen Sie das Register **Empfänger**.
3. Klicken Sie auf die Schaltfläche **Neu**.
4. Klicken Sie im Dialog **Eigenschaften Empfänger** auf die Schaltfläche **Sender** zuordnen.
5. Wählen Sie im Dialog **Sender zuordnen** den Datenbereich der gewünschten Station aus, der vom lokalen Controller empfangen werden soll.
6. Quittieren Sie die Auswahl mit **OK**.
7. Geben Sie im Eigenschaftsdialog des Empfängers Anfangsadresse des Adressbereichs ein, über den empfangen werden soll. Die Länge des Adressbereichs darf nicht geändert werden, da sie automatisch an die Länge des Sendebereichs angepasst wird. Nur wenn Sende- und Empfangsbereiche identische Längen haben, ist die Konfiguration übersetzbar!
8. Wiederholen Sie die Schritte 3 bis 7 für weitere Empfangsbereiche.
9. Für jeden zugeordneten Sender ist eine Diagnoseadresse reserviert, über die der Empfänger einen Ausfall des Senders feststellen kann.
10. Kicken Sie auf die Schaltfläche **Diagnoseadressen**, wenn Sie diese Adressen editieren wollen.
11. Bestätigen Sie Ihre Eingabe mit **OK**.

7.5 I-Device projektieren

7.5.1 PROFINET IO und I-Device

Einführung

Die direkte Kopplung z. B. von SIMATIC und SIMOTION über PROFINET war bis SIMOTION 4.0 nur mittels TCP oder UDP bzw. zusätzlicher Hardware (PN/PN-Coupler, SIMATIC-CP) möglich. Mit SIMOTION V 4.1.1.6 wurde für PROFINET IO die von PROFIBUS bekannte direkte Kopplung der Steuerungen realisiert. Über PROFIBUS kann z. B. die SIMOTION als I-Slave an die SIMATIC CPU angebunden werden. Eine vergleichbare Funktion steht auch für PROFINET IO unter der Bezeichnung I-Device zur Verfügung. Damit ist der Datenaustausch zwischen den Steuerungen über E/A-Bereiche möglich. Dadurch wird die für TCP oder UDP notwendige Programmierung der Kommunikation durch Projektierung und Systemfunktionalität ersetzt. Weiterhin entfallen die Kosten für bisher eingesetzten Hardware-Lösungen (PN/PN-Coupler, SIMATIC-CP).

Ein I-Device ist ein Controller der zusätzlich die Funktion eines IO-Devices übernimmt. Der Begriff I-Device steht für Intelligentes IO-Device. Merkmal eines Intelligenten I-Devices ist, dass dessen Ein-/Ausgangsdaten nicht unmittelbar von realen Ein-/Ausgängen dem übergeordneten IO-Controller zur Verfügung gestellt werden, sondern dass eine Vorverarbeitung der Daten im I-Device stattfindet.

Ein SIMOTION-Gerät als I-Device kann z. B. zum Datenaustausch zu einer SIMATIC-Station verwendet werden. Darüber hinaus kann beispielsweise ein SIMOTION-Gerät als I-Device auch als Anleger einer Modularen Maschine verwendet werden. Siehe dazu *Funktionsbeschreibung Motion Control Basisfunktionen für modulare Maschinen*.

Weiterhin kann ein SIMOTION-Gerät als I-Device auch für einen verteilten Gleichlauf über Projektgrenzen hinweg eingesetzt werden, siehe dazu Funktionshandbuch Motion Control Technologieobjekte Gleichlauf, Kurvenscheibe.

Hinweis

Ein I-Device kann erst ab SIMOTION V4.1.1.6 angelegt werden.

Eigenschaften eines I-Devices

Ein I-Device kann neben der Rolle eines IO-Devices an einem übergeordneten IO-Controller gleichzeitig auch ein eigenes lokales PROFINET IO-System aufspannen mit eigenen, lokalen IO-Devices, somit selbst auch ein IO-Controller sein. Beide Funktionen werden über ein und dieselbe PROFINET-Schnittstelle des Gerätes realisiert.

Das I-Device steht bei SIMOTION für PROFINET IO mit RT und mit IRT Hohe Performance zur Verfügung.

Für die Kombinationsmöglichkeit der Funktionen gibt es folgende Randbedingung:

Tabelle 7- 5 Kombinationsmöglichkeiten RT und IRT I-Device bei SIMOTION

Funktion der SIMOTION	Welche zusätzlichen Funktionen sind noch möglich			
	RT I-Device	RT-Controller	IRT I-Device	IRT-Controller
RT I-Device		X	-	X
RT-Controller	X	-	X*	X*
IRT I-Device	-	X	-	-
IRT-Controller	X	X	-	

*entweder IRT I-Device oder IRT-Controller

Die PROFINET-Schnittstelle eines I-Devices benötigt wie jedes andere IO-Device zum Betrieb Parametrierdaten. Bei einem IO-Device werden diese generell über den zugehörigen IO-Controller in Form von Parametrierdatensätzen geladen. Bei einem I-Device stehen 2 Möglichkeiten zur Verfügung. Das Interface und die Ports der PROFINET-Schnittstelle eines I-Devices können entweder vom übergeordneten IO-Controller oder lokal durch das I-Device selbst parametriert werden. Dies kann in der Konfiguration des I-Devices ausgewählt werden.

Bei der lokalen Parametrierung werden die notwendigen Daten beim Download vom Engineering-System in das I-Device geladen. Die Parametrierdaten für die PROFINET-Schnittstelle sind in den Downloaddaten für das Gerät enthalten. Der übergeordnete IO-Controller muss die PROFINET-Schnittstelle des I-Devices nicht parametrieren. Diese Möglichkeit ist zu verwenden, wenn das I-Device mit RT betrieben werden soll.

Bei der Parametrierung durch den übergeordneten IO-Controller müssen die Parametrierdaten für die PROFINET-Schnittstelle des I-Devices zusammen mit den übrigen Parametrierdaten durch den IO-Controller geladen werden. Der IO-Controller lädt dazu Parametrierdatensätze für die PROFINET-Schnittstelle in das I-Device. Wenn das I-Device mit IRT betrieben werden soll, dann müssen die Parametrierdaten durch den IO-Controller geladen werden.

Wenn das I-Device mit IRT betrieben wird, dann muss der Sendetakt des I-Devices gleich dem Sendetakt der Sync-Domain des PROFINET IO-Systems des übergeordneten IO-Controllers eingestellt werden. Wenn das I-Device mit RT betrieben wird, dann muss die Aktualisierungszeit des I-Devices gleich oder um ein Vielfaches untersetzt zum Sendetakt der Sync-Domain des PROFINET IO-Systems des übergeordneten IO-Controllers eingestellt werden.

Folgende Sendetakte und Aktualisierungszeiten in den folgenden möglichen Kombinationen müssen, wie in der Tabelle unten angegeben, eingestellt werden.

Tabelle 7- 6 Sendetakte/Aktualisierungszeiten eines I-Devices

Übergeordneter IO-Controller und I-Device mit IRT, kein lokales PROFINET IO-System oder lokales PROFINET IO-System mit IO-Devices mit RT • Sendetakt I-Device: – muss gleich zum Sendetakt des übergeordneten IO-Controllers sein. – einzustellen am I-Device in Eigenschaften <Profinet Interface> unter dem Reiter PROFINET in der Drop-Down Box "Sendetakt"
Übergeordneter IO-Controller mit IRT und I-Device mit RT, lokales PROFINET IO-System mit IRT • Aktualisierungszeit I-Device: – muss ein ganzzahliges Vielfaches des Sendetaktes des übergeordneten IO-Controllers und des Sendetaktes des IO-Controllers im I-Device sein – einzustellen am I-Device-Stellvertreter in Eigenschaften <Profinet Interface> im Reiter IO-Zyklus unter Aktualisierungszeit
Übergeordneter IO-Controller und I-Device mit RT, kein lokales PROFINET IO-System • Aktualisierungszeit I-Device: – Es können die für das I-Device möglichen Aktualisierungszeiten eingestellt werden – einzustellen am I-Device-Stellvertreter in Eigenschaften <Profinet Interface> im Reiter IO-Zyklus unter Aktualisierungszeit
Übergeordneter IO-Controller und I-Device mit RT, lokales PROFINET IO-System mit IRT: • Aktualisierungszeit I-Device: – muss gleich oder kleiner als der Sendetakt des IO-Controllers im I-Device sein. – einzustellen am I-Device-Stellvertreter in Eigenschaften <Profinet Interface> im Reiter IO-Zyklus unter Aktualisierungszeit

Das folgende Bild zeigt, wie ein I-Device an einem übergeordneten IO-Controller projektiert werden kann. Der übergeordnete IO-Controller spannt ein PROFINET IO-System auf, in dem sich das I-Device befindet. Das I-Device kann ein lokales PROFINET IO-System aufspannen. Jedes dieser PROFINET IO-Systeme kann einer eigenen Sync-Domain angehören. Das I-Device darf aber nur einer der möglichen Sync-Domains zugeteilt werden, da eine PROFINET-Schnittstelle nur genau einer Sync-Domain angehören kann.

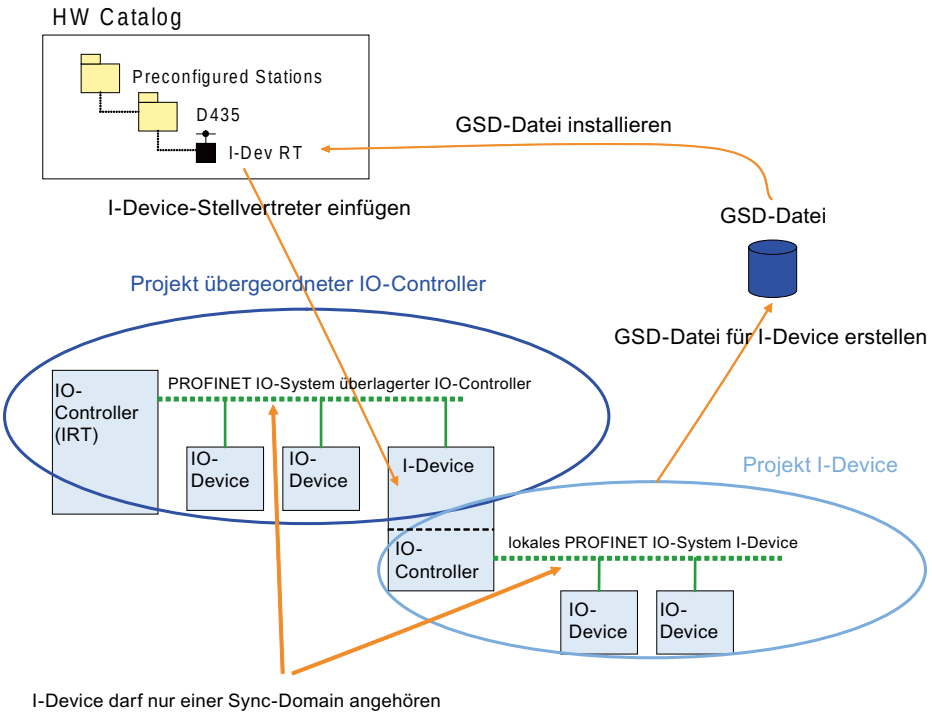


Bild 7-27 Projektierung I-Device

Vorgehen beim Projektieren

- Das I-Device selbst und der IO-Controller, an dem es betrieben werden soll, sollte in verschiedenen Projekten angelegt werden.
- Für das I-Device muss der I-Device Modus der PROFINET-Schnittstelle eingeschaltet werden. Weiterhin müssen die Ein- und Ausgangsbereiche im I-Device für den Datenaustausch mit dem übergeordneten IO-Controller konfiguriert werden.
- Nach dem Anlegen und Projektieren eines I-Devices muss eine GSD-Datei für dessen I-Device-Stellvertreter erstellt und installiert werden. Der I-Device-Stellvertreter steht danach im Hardware Katalog unter Preconfigured Stations zur Verfügung.
- Im Anschluss muss der I-Device-Stellvertreter aus dem Hardware Katalog aus Preconfigured Stations in das PROFINET IO-System des übergeordneten IO-Controllers eingefügt werden.

Da ein I-Device-Stellvertreter nur über einen manuellen Vorgang im Hardware Katalog erzeugt werden kann, findet damit kein automatischer Abgleich zwischen dem Projekt mit dem I-Device und dem entsprechenden I-Device-Stellvertreter in der GSD-Datei statt. Folglich darf die Konfiguration des I-Devices nachträglich nicht mehr verändert werden. Wenn jedoch eine Änderung stattfindet, dann bedingt dies eine neue GSD-Datei, die wieder erzeugt und installiert werden muss. Wenn die Konfiguration eines bestimmten I-Devices mehrmals nachträglich geändert und mehrmals von diesem I-Device eine GSD-Datei erstellt und installiert wird, erscheint immer die neueste Version im Hardware Katalog unter Preconfigured Stations. Um das Erneuern der Version sicherzustellen, muss jedoch der beim Erstellen und Installieren der GSD-Datei verwendete Bezeichner für den I-Device-Stellvertreter identisch sein. Nur die Ein- bzw. Ausgangsadressen für den Datenaustausch dürfen im Projekt des übergeordneten IO-Controllers geändert werden.

Da I-Devices an ihren übergeordneten IO-Controller und die am PROFINET IO-System eines einzelnen I-Devices angeschlossenen IO-Devices über ein und dieselbe PROFINET-Schnittstelle angeschlossen werden, befinden sich alle die genannten Geräte auch in ein und demselben Ethernet-Subnetz. Dies hat zur Konsequenz, dass die Gerätenamen und IP-Adressen aller dieser Geräte voneinander verschieden bzw. die Subnetzmasken identisch sein müssen. Diesem Fakt ist dann besondere Aufmerksamkeit zu schenken, wenn sich der übergeordnete IO-Controller und das I-Device in verschiedenen Projekten befinden. Denn HW Konfig kann die Konsistenz der Gerätenamen, IP-Adressen und Subnetzmasken über verschiedene Projekte hinweg nicht prüfen.

Gerätename (NameOfStation) für I-Device

Wie für alle IO-Devices an PROFINET IO muss auch für ein I-Device ein Gerätenamen in der Projektierung festgelegt werden. Der Gerätename (NameOfStation) für das I-Device wird in den Eigenschaften von dessen PROFINET-Schnittstelle eingestellt und ist damit identisch mit dem Gerätenamen des IO-Controllers im I-Device. Dieser eingestellte Name wird beim Erstellen und Installieren der GSD-Datei für den I-Device-Stellvertreter mit in die GSD-Datei geschrieben. Beim Einfügen des I-Device-Stellvertreters in das PROFINET IO-System des übergeordneten IO-Controllers wird der in der GSD-Datei vorbelegte Gerätename übernommen in die Konfiguration. Auf jeden Fall muss sichergestellt werden, dass der Gerätename in der Konfiguration des übergeordneten IO-Controllers identisch ist zum Gerätenamen, der für das I-Device festgelegt ist. Folglich darf der Gerätenamen nach dem Hinzufügen in das PROFINET IO-System des übergeordneten IO-Controllers nicht mehr geändert werden.

Wenn die Gerätenamen verschieden sind, dann kann der übergeordnete IO-Controller das betreffende I-Device nicht hochfahren und in folge dessen nicht den zyklischen Austausch der Ein-/Ausgangsdaten starten.

Folgende Fälle führen zu verschiedenen Gerätenamen und müssen deshalb vermieden werden:

- Wenn Sie als übergeordneten IO-Controller ein SIMOTION-Gerät verwenden, darf der Gerätename (NameOfStation) des I-Device keine "-" enthalten. Denn diese werden in "x" umgewandelt, wenn das I-Device in das PROFINET IO-System eingefügt wird.
- Da ein Gerätename innerhalb eines Ethernet-Subnetzes nicht zweimal vorkommen darf, wird beim Einfügen eines I-Device-Stellvertreters in das PROFINET IO-System seines übergeordneten IO-Controllers der Gerätename geändert, wenn dieser schon vorhanden ist. Aus diesem Grund muss dafür gesorgt werden, dass der in der GSD-Datei vorbelegte Gerätenamen noch nicht verwendet wird.
- Wenn mehr als ein I-Device-Stellvertreter aus ein und demselben Eintrag von Preconfigured Stations in das PROFINET IO-System des übergeordneten IO-Controllers eingefügt wird, dann wird der in der GSD-Datei vorbelegte Gerätenamen verändert. Deshalb muss für jedes I-Device, welches in einem PROFINET IO-System verwendet werden soll, auch je ein I-Device-Stellvertreter angelegt werden.

7.5.2 I-Device anlegen

Voraussetzung

Sie haben in HW Konfig (SIMATIC Manager oder SIMOTION SCOUT) bereits ein Projekt angelegt und eine Station mit Rack bzw. einen SIMOTION Controller angelegt. Sie haben das PROFINET IO-System bereits projektiert und möchten nun das I-Device projektieren.

Hinweis

Beachten Sie bei der Projektierung des I-Device die möglichen Einstellungen zur RT-Klasse, siehe PROFINET I-Device

Vorgehensweise

1. Doppelklicken Sie auf das Interface-Modul der CPU. Der Dialog Eigenschaften öffnet sich.
2. Wählen Sie das Register Allgemein aus und ändern Sie ggf. den Gerätenamen (keine "-").

   Bild Dialog Reiter Allgemein
3. Wählen Sie das Register I-Device aus.

   Bild Dialog Reiter I-Device
4. Wählen Sie den I-Device Modus aus.
5. Wählen Sie aus, ob die Parametrierung der PN-Schnittstelle und deren Ports am überlagerten IO-Controller erfolgen soll. Diese Option muss immer dann angewählt werden, wenn die zyklische Kommunikation zwischen dem übergeordneten IO-Controller und dem I-Device mit IRT laufen soll. Damit werden dann auch Ports in der GSD-Datei angelegt und beim Anlauf werden die Parametrierdatensätze in den Controller des I-Device geladen. Wenn diese Option nicht ausgewählt wird, kann die zyklische Kommunikation zwischen dem übergeordneten IO-Controller und den I-Devices nur über RT erfolgen.
6. Wählen Sie **I-Device/Applikation taktsynchron betreiben** aus, wenn Sie die Kommunikation mit IRT realisieren wollen. Am I-Device-Stellvertreter wird damit im Dialog Eigenschaften der PROFINET-Schnittstelle des I-Device-Stellvertreters zusätzlich das Register **Applikation** eingeblendet, in dem dann **I-Device/Applikation taktsynchron betreiben** ausgewählt werden kann, um das I-Device taktsynchron betreiben zu können. Wenn ein I-Device mit IRT betrieben werden soll, dann muss sowohl Parametrierung der PN-Schnittstelle und deren Ports am überlagerten IO-Controller als auch **I-Device/Applikation taktsynchron betreiben** gesetzt sein.
7. Wenn das I-Device mit IRT betrieben wird, dann muss dessen Sendetakt eingestellt werden. Wählen Sie dazu das Register **PROFINET** aus und stellen Sie einen entsprechenden Sendetakt ein.
8. Klicken Sie jeweils **Neu**..., um die virtuellen Subslots (Ein- und Ausgangs-Adresse) anzulegen, und konfigurieren Sie diese gemäß den Anforderungen. Damit konfigurieren Sie den E/A-Bereich des I-Device, über den Daten mit dem überlagerten IO-Controller ausgetauscht werden. In den Registern **Sender** und **Empfänger** müssen Sie keine Einstellungen mehr durchführen.

9. Klicken Sie **OK**, um die Einstellungen zu übernehmen und speichern Sie das Projekt.
10. Fahren Sie fort mit I-Device-Stellvertreter erstellen.

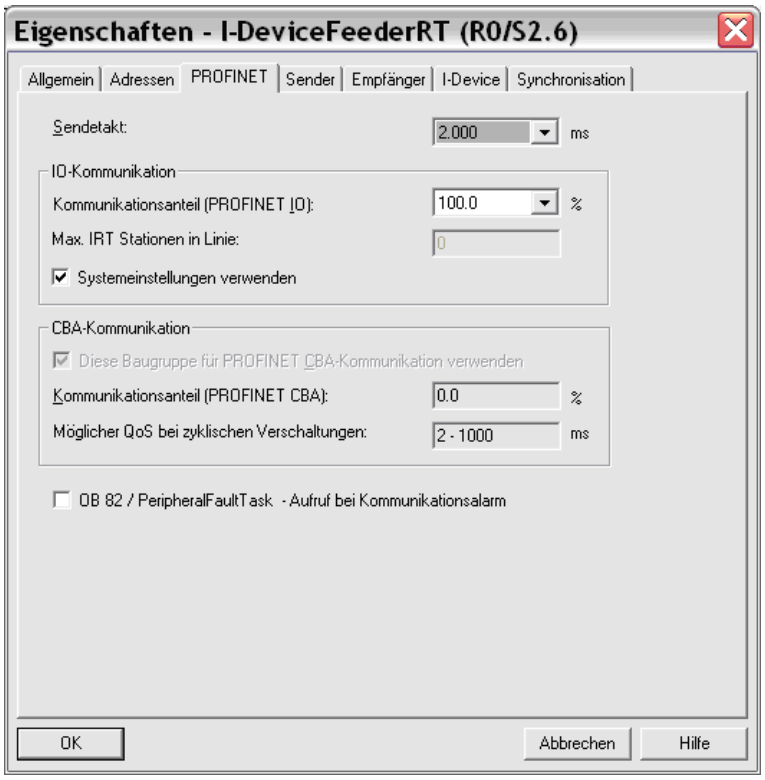


Bild 7-28 Einstellung Sendetakt I-Device

7.5.3 GSD-Datei für I-Device exportieren

Das Exportieren der GSD-Datei wird immer dann benötigt, um ein I-Device in einem Projekt auf einem anderen PC verwenden zu können.

Voraussetzung

Sie haben bereits die Baugruppe, die als I-Device verwendet werden soll, projektiert.

1. Speichern Sie zuerst das Projekt.
2. Führen Sie **Extras > GSD-Datei für I-Device erstellen...** aus. Der Dialog GSD-Datei für I-Device erstellen wird eingeblendet.

3. Wählen Sie das I-Device aus und geben Sie eine Bezeichnung für den I-Device-Stellvertreter ein. Unter diesem Bezeichner erscheint der I-Device-Stellvertreter unterhalb Preconfigured Stations im HW-Katalog.

4. Klicken Sie auf **Erstellen** und anschließend auf **Exportieren**. Der Dialog **Ordner suchen** wird eingeblendet.
5. Wählen Sie den Pfad aus, in dem die GSD-Datei des I-Device-Stellvertreters abgelegt werden soll und klicken Sie auf **OK.**

7.5.4 I-Device-Stellvertreter erstellen

Für das Erstellen eines I-Device-Stellvertreters gibt es 2 Möglichkeiten. Zum einen kann dieser mit Hilfe einer vorher exportierten GSD-Datei über **Extras > GSD-Dateien installieren ...** und zum anderen im Dialog **GSD-Datei für I-Device erstellen** erzeugt werden.

Voraussetzung

Sie haben bereits die Baugruppe, die als I-Device verwendet werden soll, projektiert und von dieser die GSD-Datei exportiert.

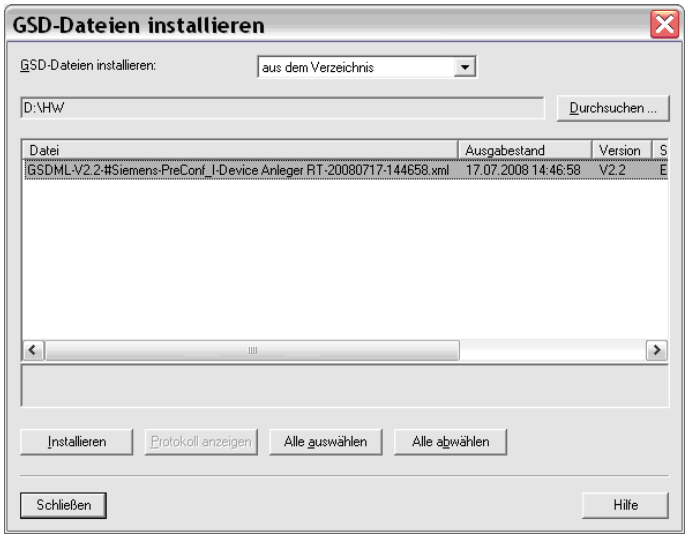
Vorgehensweise 1

1. Speichern Sie zuerst das Projekt.
2. Erstellen Sie wie in Kapitel GSD-Datei für I-Device exportieren (Seite 208) beschrieben ein I-Device. Anstatt es zu exportieren, wird die Datei sofort installiert.
3. Klicken Sie auf **Erstellen** und anschließend auf **Installieren.**

4. Klicken Sie auf **Schließen**. Anschließend steht der I-Device-Stellvertreter unterhalb von Preconfigured Stations zur Verfügung.

Vorgehensweise 2

1. Führen Sie **Extras > GSD-Dateien...** aus. Der Dialog GSD-Dateien insatllieren wird eingeblendet.
2. Klicken Sie auf **Durchsuchen...** Der Dialog **Ordner suchen** wird eingeblendet.
3. Wählen Sie den Pfad aus, in dem GSD-Dateien für I-Device-Stellvertreters abgelegt ist und klicken Sie auf **OK.**



4. Wählen Sie die gewünschten GSD-Dateien aus und klicken Sie auf **Installieren**.
5. Klicken Sie auf **Schließen**. Anschließend stehen die I-Device-Stellvertreter unterhalb von Preconfigured Stations zur Verfügung.

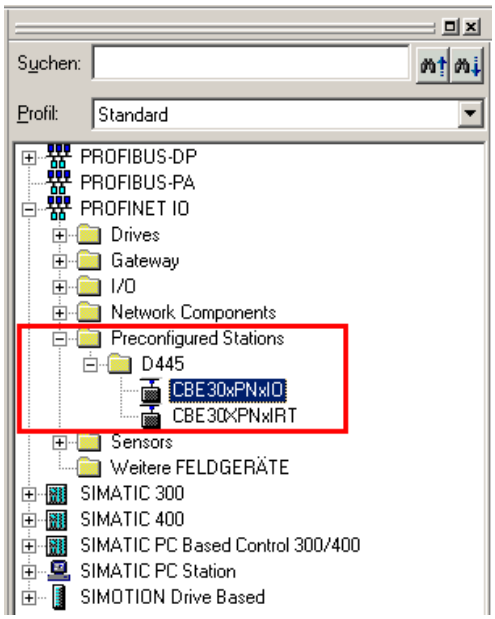


Bild 7-29 i-Device Eintrag im Hardware Katalog

7.5.5 I-Device-Stellvertreter am übergeordneten IO-Controller einfügen

Voraussetzung

Sie haben bereits einen I-Device-Stellvertreter erstellt. Ein Projekt ist geöffnet und ein IO-Controller mit einem PROFINET IO-System ist bereits projektiert.

I-Device-Stellvertreter einfügen

1. Öffnen Sie den Hardware-Katalog.
2. Ziehen Sie den entsprechenden I-Device-Stellvertreter vom Hardware-Katalog (PROFINET IO > Preconfigured Stations) auf das PROFINET IO-System. Der I-Device-Stellvertreter wird am PROFINET IO-System wie ein normales IO-Device angezeigt. Abhängig davon, ob **Parametrierung der PN-Schnittstelle und deren Ports am überlagerten IO-Controller** angewählt ist, werden Ports angezeigt oder nicht.

Folgendes Bild zeigt einen übergeordneten IO-Controller mit einem I-Device mit RT.

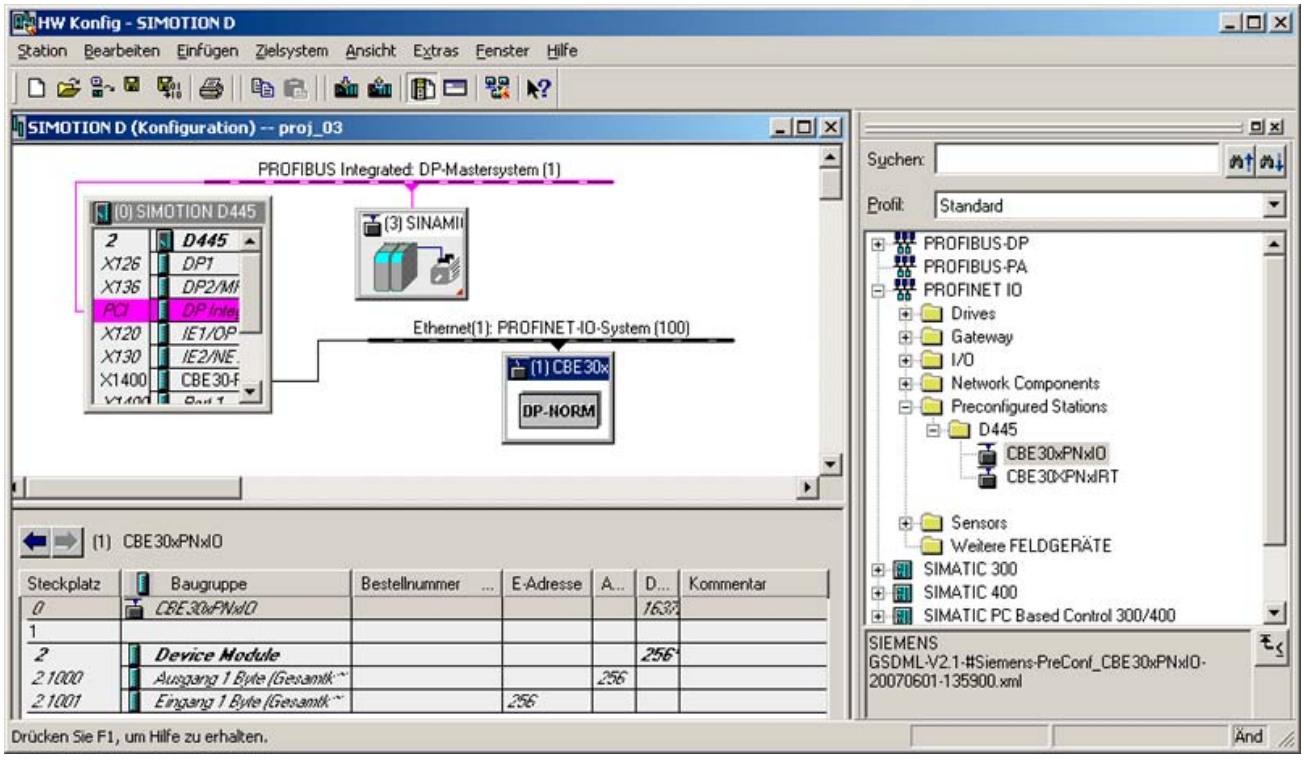


Bild 7-30 RT I-Device am IO-Controller

Folgendes Bild zeigt einen übergeordneten IO-Controller mit einem I-Device mit IRT.

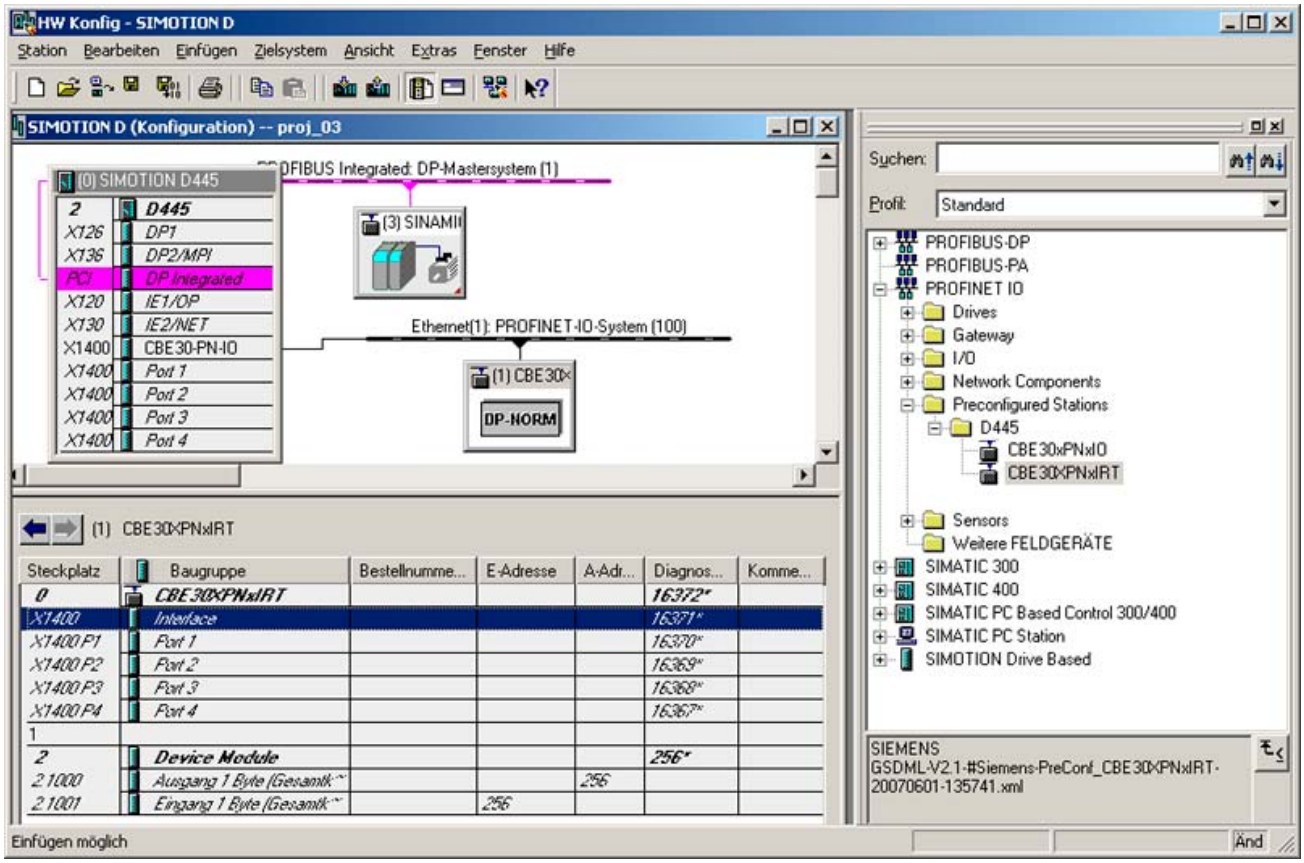


Bild 7-31 i-Device am IO-Controller

Die Anzahl der Submodule entspricht der Anzahl der projektierten Submodule des I-Devices in der GSD-Datei. Modul und Submodule (virtuellen Subslots) sind nicht löschbar.

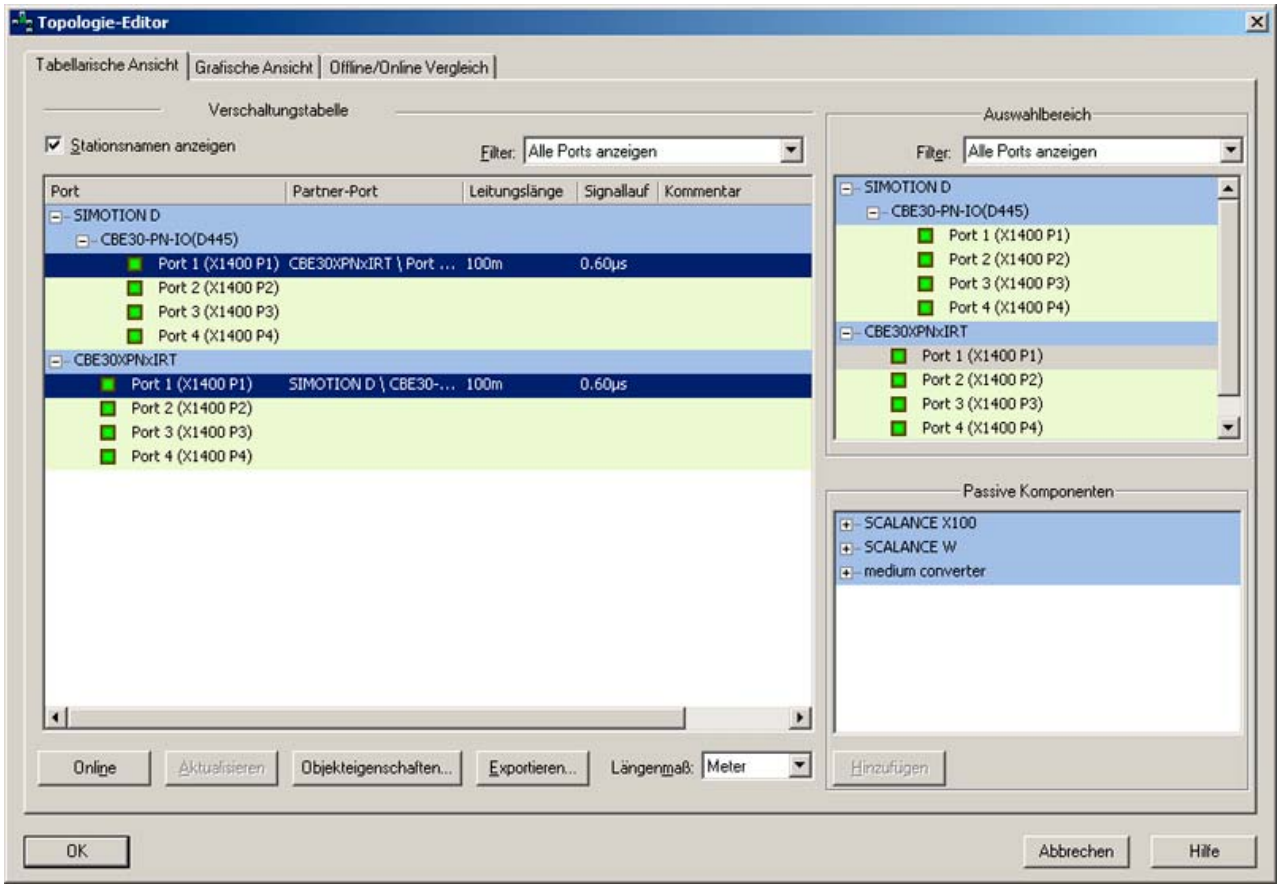


Bild 7-32 I-Device Ports verschalten

IP-Adresse für I-Device-Stellvertreter vergeben

1. Doppelklicken Sie auf das I-Device um den **Eigenschaften** Dialog aufzurufen.
2. Wählen Sie die Option **IP-Adresse durch IO-Controller zuweisen** ab.

   Die IP-Adresse sollte nicht vom übergeordneten IO-Controller zugewiesen werden, da diese bereits im Step7-Projekt des I-Device vergeben wird.

Synchronisationsart und Taktsynchronität einstellen für I-Device mit IRT

1. Doppelklicken Sie auf den Interface-Eintrag (X1400) um den Dialog **Eigenschaften Interface** aufzurufen.
2. Wählen Sie im Register **Synchronisation** als Synchronisationsart **Sync-Slave** und als RT-Klasse **IRT** aus.

   Erst dann werden die Felder im Register **Applikation** aktiv.
3. Wählen Sie im Register **Applikation** die Option **IO-Device/Applikation taktsynchron betreiben** aus und konfigurieren Sie die Applikation entsprechend Ihren Anforderungen.

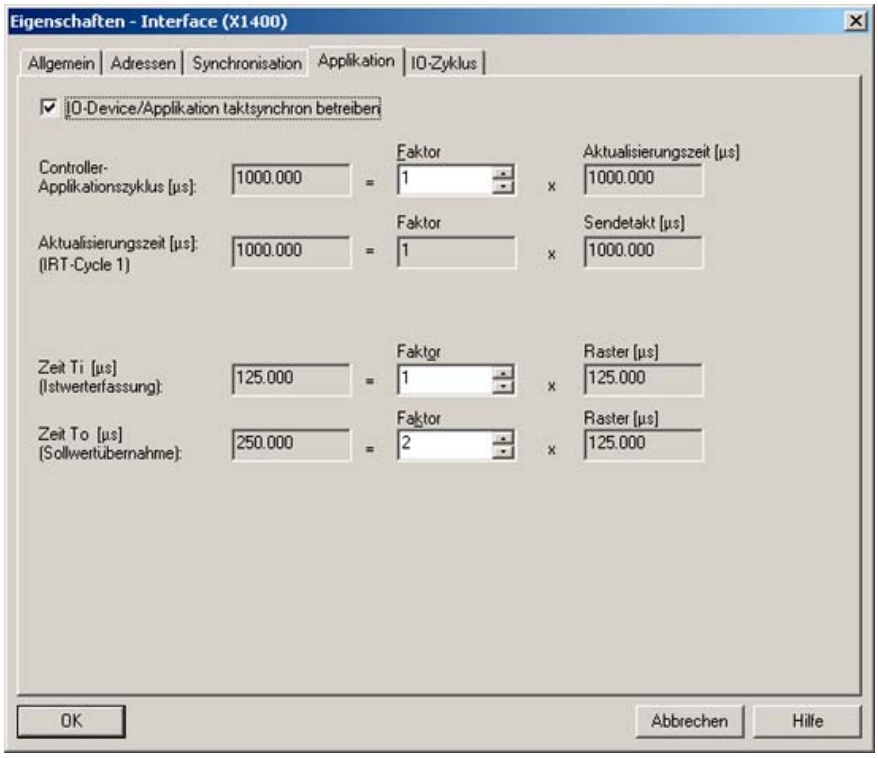


Bild 7-33 i-Device Applikation

Aktualisierungszeit und Sendetakt einstellen

I-Device mit RT

- Für I-Devices mit RT ist die Aktualisierungszeit einzustellen. Doppelklicken Sie dazu auf das PROFINET IO System und wählen Sie im Dialog **Eigenschaften PROFINET Subnetz** das Register **Aktualisierungszeit** aus. Stellen Sie dort die Aktualisierungszeit ein.

I-Device mit IRT

- Für I-Devices mit IRT ist der Sendetakt einzustellen. Der Sendetakt im Projekt des I-Devices muss gleich zum Sendetakt im Projekt des übergeordneten IO-Controllers eingestellt werden. Den Sendetakt des übergeordneten Projektes können Sie in HW Konfig über **Bearbeiten > PROFINET IO > Domain Management** einstellen.

7.5.6 I-Device-Stellvertreter löschen

Die GSD-Dateien für die I-Device-Stellvertreter unter Preconfigured Stations befinden sich in folgendem Verzeichnis:

<Program Files>\Siemens\Step7\S7DATA\GSD,
z. B. GSDML-V2.2-#Siemens-PreConf_**I-Dev RT**-20080704-095947.xml.

I-Dev RT ist dabei der Name des I-Device-Stellvertreters. Die I-Device-Stellvertreter können gelöscht werden, indem die betreffenden XML-Files gelöscht werden. Die Anzeige der I-Device-Stellvertreter unterhalb von Preconfigured Stations wird erst aktualisiert, wenn erneut eine GSD-Datei erstellt und installiert wird.

7.6 Kommunikationsprojektierung laden

7.6.1 PROFINET IO Projektierung laden

Voraussetzung

Es ist ein PG/PC angeschlossen, mit dem Sie ONLINE gehen können.

Vorgehensweise

Die Projektierungsdaten müssen nach erfolgter Projektierung von PROFINET IO in alle beteiligten Controller geladen werden.

1. Markieren Sie in NetPro das Ethernet-Subnetz und wählen den Menübefehl **Zielsystem > Laden im aktuellen Projekt > Stationen am Subnetz**.

7.7 Datenaustausch zwischen SIMATIC und SIMOTION über PROFINET

7.7.1 Datenaustausch durch Verwenden von I-Devices

Beschreibung

Die Kopplung zwischen SIMATIC und SIMOTION ist mit folgenden Funktionen möglich:

- TCP/ UDP*)-Anwenderkommunikation
- PROFINET IO/ RT, über S7-300 CP als Device
- PROFINET IO/ RT, über PN/PN Koppler
- PROFINET IO/ RT, über I-Device

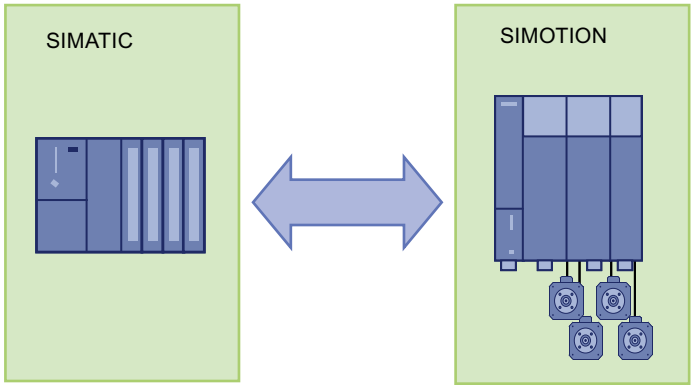


Bild 7-34 Datenaustausch zwischen SIMOTION und SIMATIC

Weitere Informationen

In den SIMOTION Utilities & Applications ist ein FAQ zum Thema PROFINET RT I-Device-Kopplung zwischen einer SIMOTION Steuerung und einer SIMATIC Steuerung beschrieben. Die SIMOTION Utilities & Applications sind im Lieferumfang von SIMOTION SCOUT enthalten.

Die folgenden drei Anwendungsfälle werden betrachtet:

Fall A: SIMOTION und SIMATIC als I-Device in einem Projekt, SIMOTION als Controller im zweiten separaten Projekt.

Fall B: Ein Projekt für alle Komponenten.

Fall C: Mehrfachverwendung eines I-Devices.

Nähere Informationen zur Projektierung von I-Devices, siehe auch das Kapitel I-Device projektieren (Seite 202)

Beachten Sie auch den FAQ zur Projektierung FAQ I-Device (http://support.automation.siemens.com/WW/view/de/29578823)

7.7.2 PN-PN-Coupler

Beschreibung

Der PN/PN Coupler dient dazu, zwei PROFINET IO Systeme miteinander zu verbinden und Daten zwischen diesen auszutauschen. Die maximale Größe der übertragbaren Daten beträgt 256 Bytes Eingangsdaten und 256 Bytes Ausgangsdaten.

Der PN/PN Coupler hat als ein Gerät zwei PROFINET-Schnittstellen, die jeweils mit einem anderen Subnetz verbunden werden.

Hinweis

Der PN/PN-Coupler kann nur als Gerät mit der RT-Klasse **RT** eingesetzt werden.

In der Projektierung werden aus einem PN/PN Coupler zwei IO-Devices abgeleitet, und zwar für jeden Controller mit ihrem Subnetz jeweils ein IO-Device. Der jeweils andere Teil des PN/PN Couplers wird als Koppel-Partner bezeichnet. Mit Abschluss der Projektierung werden die beiden Teile zusammengeführt.

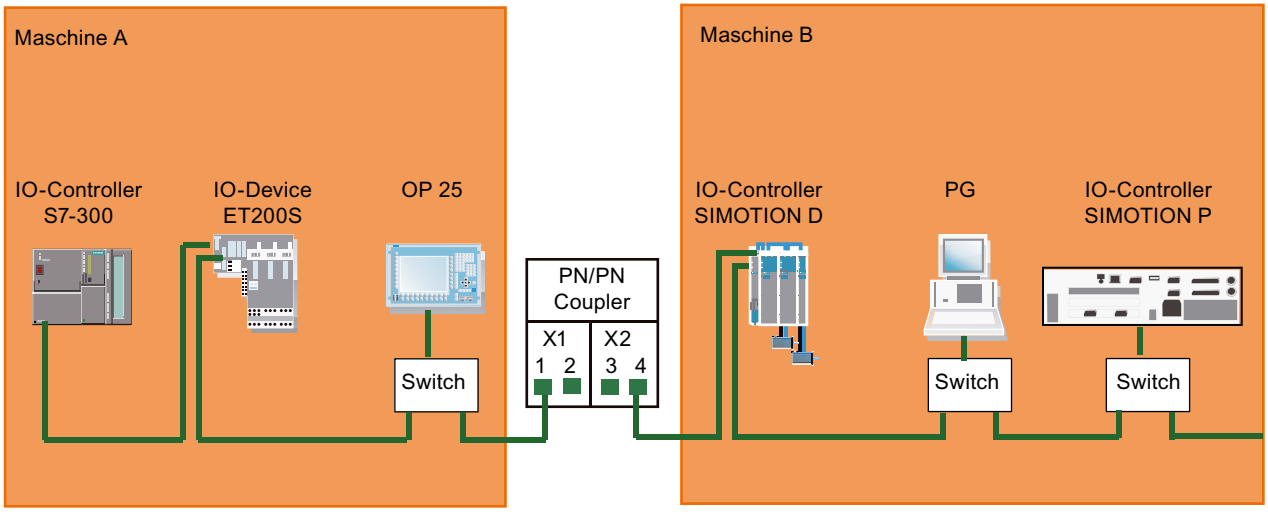


Bild 7-35 Kopplung zweier PROFINET-Subnetze mit einem PN/PN-Coupler

Die beiden Maschinen im Bild sind über den PN/PN-Coupler galvanisch getrennt. Soll es z. B. auch möglich sein, mit dem PG von Maschine B auf Maschine A online zu gehen, kann im PN/PN-Coupler eine Brücke zwischen Port 2 und 3 gesteckt werden. Damit geht die galvanische Trennung verloren.

Hinweis

Detaillierte Information über PN/PN-Coupler finden Sie in der entsprechenden Gerätedokumentation.

PN/PN-Coupler projektieren

Für die Projektierung des PN/PN-Couplers werden zwei PROFINET-Geräte angelegt, die linke (X1) und die rechte (X2) Seite. Den PN/PN-Coupler projektieren Sie mit HW Konfig. Wenn beide Subnetze in einem Projekt projektiert sind, können Sie mit STEP 7 den PN/PN-Coupler für beide Subnetze projektieren. Sind die Subnetze in verschiedenen Projekten projektiert, müssen Sie den Coupler in jedem Projekt projektieren.

Hinweis

Der PN/PN-Coupler wird zum Datenaustausch SIMOTION mit SIMATIC verwendet. Er ist aber auch für den schnellen Austausch von F-Signalen zwischen SIMATIC-CPU die bevorzugte Lösung.

7.7.3 Kommunikation über Standardprotokolle

Beschreibung

Mit TCP und UDP können Daten zwischen SIMOTION, SIMATIC, anderen Steuerungen und Fremdsystemen ausgetauscht werden. Die PROFINET Schnittstelle ist eine Standard-Ethernet Schnittstelle und unterstützt diese Protokolle. Das Anwenderprogramm muss dabei die Verwaltung der Kommunikationsverbindung realisieren. Über diese Verbindung können Sie z. B. Daten zwischen einem SIMATIC CPU und SIMOTION Controller über PROFINET austauschen.

UDP ist ein verbindungsloses Protokoll, d. h. es wird gesendet, ohne Rückmeldung ob der Empfänger die Nachricht erhalten hat. TCP ist ein verbindungsortientiertes Protokoll, d. h. es wird zuerst eine logische Verbindung aufgebaut. Erst wenn diese Verbindung steht wird gesendet. Die Verbindung d. h. der Empfang der Nachrichten wird überwacht.

Datenaustausch über TCP:

- Verbindungsaufbau
- Datenverwaltung
- Verbindungsüberwachung
- Verbindungsabbau

Die Verwendung der Systembefehle ist ausführlich im Abschnitt **Ethernet Allgemein (TCP/IP- und UDP-Verbindungen)** beschrieben.

Siehe auch

Funktion _tcpOpenServer (Seite 132)
Funktion _tcpOpenClient (Seite 132)
Funktion _tcpReceive (Seite 133)
Funktion _tcpSend (Seite 133)
Funktion _tcpCloseConnection (Seite 134)
Funktion _tcpCloseServer (Seite 134)
Funktion _udpSend (Seite 134)
Funktion _udpReceive (Seite 135)

7.8 Diagnose und Alarmverhalten

7.8.1 PROFINET IO Alarm- und Diagnosemeldungen an SIMOTION

Beschreibung

Für PROFINET IO ist eine Alarm- und Diagnosefunktionalität für die PROFINET-Geräte verfügbar.

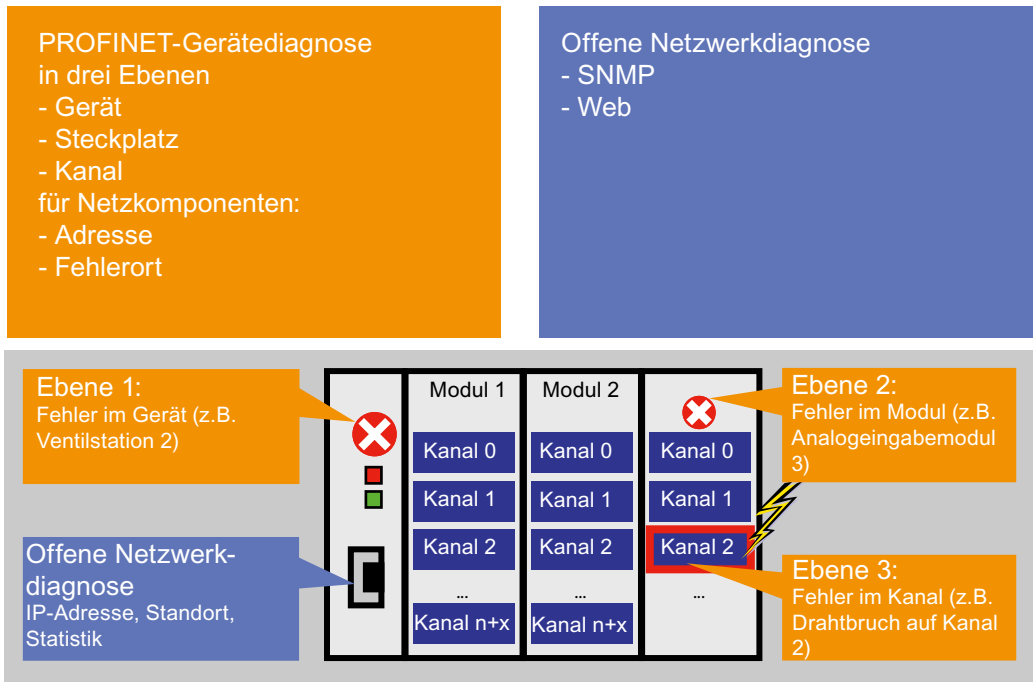


Bild 7-36 Diagnoseübersicht

Gerätediagnose

Die Gerätediagnose kann in drei Ebenen aufgeteilt werden. Detaillierte Informationen finden Sie im Abschnitt Diagnosemodell.

7.8.2 Diagnosemodell

Bei PROFIBUS DP wird für Diagnose- und Statusmeldungen eines Slaves ein Diagnosetelegramm zum Master übertragen. Ein Diagnosetelegramm enthält den gesamten Diagnose-Status eines Slaves. In einem DP-Diagnosetelegramm ist nur die Abbildung von Parametrier- und Konfigurationsfehlern standardisiert. Weitere Diagnose- und Statusmeldungen können ergänzt werden, diese werden jedoch herstellerspezifisch kodiert.

PROFINET IO setzt von Anfang an auf vollständig standardisierte Diagnosemechanismen. Dies ist für eine herstellerübergreifende Geräte- und System-Diagnose äußerst hilfreich.

Aufgrund der größeren Mengengerüste ist es nicht möglich, die Statusinformationen aller Stationen im IO-Controller zu halten. Deshalb werden nur aktuelle Diagnose-Ereignisse über standardisierte Alarme an den IO-Controller übertragen.

Durch die Nutzung eines quittierten Dienstes wird die Übertragung der Diagnoseereignisse in kausaler Reihenfolge ermöglicht. Der Status einer Station wird von dieser gespeichert und kann von einem Diagnosesystem jederzeit und direkt über standardisierte Datensätze ausgelesen werden, siehe entsprechende Dokumentation zu STEP7.

Zugriff auf Alarm- und Diagnosedaten

Für PROFINET IO wird zwischen folgenden Alarm- bzw. Diagnosemeldungen unterschieden:

- Alarme, die von IO-Devices an den IO-Controller geschickt werden
- Alarme, die im IO-Controller auftreten

Die folgende Grafik zeigt die Zugriffsmöglichkeiten auf Diagnosedaten:

1. Diagnose auf PG
   Das PG liest die Diagnose direkt vom IO-Device. Die Visualisierung findet im PG statt.
2. Diagnose in der Steuerung
   Das IO-Device sendet die Diagnose an den IO-Controller, die Reaktion auf die Störung findet in der Steuerung statt.

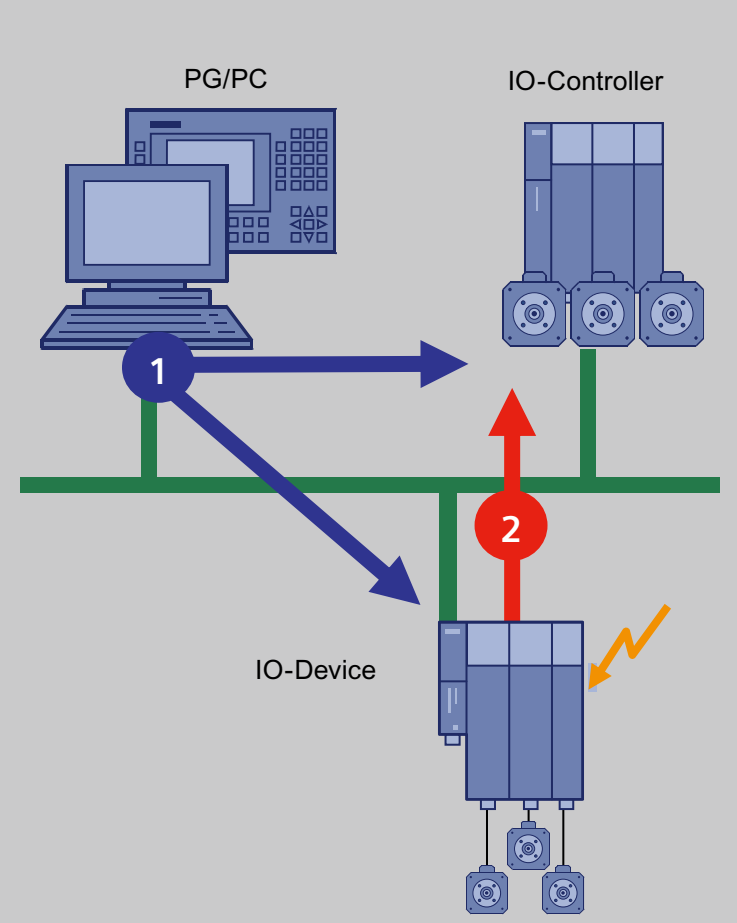


Bild 7-37 Zugriff auf Diagnosedaten

7.8.3 Alarme am IO-Controller

Beschreibung

Im IO-Controller werden eine Reihe von Alarmen ausgegeben. Die auftretenden Alarme werden mit der betreffenden EventID im Diagnosepuffer von SIMOTION aufgelistet. Folgende Alarme können unterschieden werden:

- Alarme beim direktem Datenaustausch zwischen IO-Controllern
- Stationsalarme, die von der PROFINET-Schnittstelle gemeldet werden

Folgende Tabelle zeigt die Alarme von PROFINET IO wie sie in SIMOTION abgebildet werden:

Alarm (TSI#InterruptId)	TSI#eventClass	TSI#faultId	Bedeutung
Stationsausfall (_SC_STATION_DISCONNECTED (= 202))	16#39	16#CA	Systemfehler PROFINET IO; Hierfür gibt es nur ein kommendes Ereignis, ein gehendes Ereignis wird auf 16#38 - 16#CB abgebildet und zwar für jedes vorhandene IO-Device.
		16#CB	Stationsausfall eines IO Device
		16#CC	IO Device gestört. Kanaldiagnose oder herstellerspezifische Diagnose steht an.
Stationswiederkehr (_SC_STATION_RECONNECTED (= 203))	16#38	16#CB	Stationswiederkehr eines IO Device ohne Fehler
		16#CC	Störung der IO Device behoben
		16#CD	Stationswiederkehr eines IO Device, aber Fehler: Sollaufbau <> Istaubau
		16#CE	Stationswiederkehr eines IO Device, aber Fehler bei Baugruppenparametrierung

Verwendung der TaksStartInfo

Informationen zum Verwenden der TaskStartInfo für die PeripheralFaultTask finden Sie im Handbuch **Basisfunktionen**.

7.8.4 Alarme vom IO-Device an den IO-Controller

Beschreibung

Die Alarme werden über den PROFINET-Alarmmechanismus vom IO-Device zu seinem zugehörigen IO-Controller übertragen. Die Alarme werden in den Diagnosepuffer eingetragen und können über die PeripheralFaultTask ausgewertet werden. Folgende Tabelle zeigt wie die Alarme auf die PeripheralFaultTask abgebildet werden.

Alarm (TSI#InterruptId)	TSI#event Class	TSI#faultId	Bedeutung
Diagnosis (kommend)	16#39	16#42	Kommender Diagnosealarm.
Diagnosis disappears (gehend) Multicast Communication Mismatch Port Data Change Notification Sync Data Changed Notification Isochronous Mode Problem Notification Network component problem notification (_SC_DIAGNOSTIC_INTERRUPT (=201))	16#38	16#42	Gehender Diagnosealarm
Prozessalarm (_SC_PROCESS_INTERRUPT (= 200))	16#11	16#41	Prozessalarm
Pull Alarm Plug Alarm	16#39	16#51	PROFINET IO Modul wurde entfernt oder kann nicht adressiert worden.

Alarm (TSI#InterruptId)	TSI#event Class	TSI#faultId	Bedeutung
Plug Wrong Submodule Alarm Return of Submodule Alarm (_SC_PULL_PLUG_INTERRUPT (=216))		16#54	PROFINET IO Submodul wurde entfernt oder kann nicht adressiert worden.
	16#38	16#54	PROFINET IO Modul oder Submodul wurde gesteckt, Modultyp OK (Istaufbau = Sollaufbau)
		16#55	PROFINET IO Modul oder Submodul wurde gesteckt, aber falscher Modultyp (Istaufbau <> Sollaufbau)
		16#56	PROFINET IO Modul oder Submodul wurde gesteckt, aber Fehler bei der Baugruppenparameterierung
		16#58	IO Status eines Modul hat sich von BAD nach GOOD geändert
Status			Nicht unterstützt
Update			Nicht unterstützt
Time data changed notification			Nicht unterstützt
Upload and storage notification			Nicht unterstützt
Pull module			Nicht unterstützt
Manufacturer specific			Nicht unterstützt
Profile specific			Nicht unterstützt

Alarmtypen, die mit "Nicht unterstützt" gekennzeichnet sind, werden vom SIMOTION - Controller mit "not supported" quittiert und nicht in den Diagnosepuffer eingetragen.

Verwendung der TaksStartInfo

Informationen zum Verwenden der TaskStartInfo für die PeripheralFaultTask finden Sie im Handbuch **Basisfunktionen**.

Diagnosedaten übertragen

Der genaue Grund für den Alarm wird in Form von Diagnosedaten bereit gestellt. Diese können über die Funktion _readDiagnosticData ausgelesen werden. Der Länge ist auf 255 Byte beschränkt.

7.8.5 Alarme bei direktem Datenaustausch zwischen IO-Controllern

Beschreibung

Bei PROFINET IO mit IRT gibt es eine Kommunikationsüberwachung zwischen IO-Controllern. Wenn diese feststellt, dass keine IRT-Daten mehr empfangen werden - entweder es kommen überhaupt keine Daten oder kommen zu spät - wird ein Stationsausfallalarm generiert. Wenn eine Wiederaufnahme der Kommunikation festgestellt wird, so wird ein Stationswiederkehralarm generiert. Kommen IRT-Daten dreimal zu spät, wird ein Stationsausfallalarm gemeldet.

Folgende Tabelle zeigt die Alarme von PROFINET IO zwischen IO-Controllern des direkten Datenaustauschs wie sie in SIMOTION abgebildet werden:

Alarm (TSI#InterruptId)	TSI#eventClass	TSI#faultId	Bedeutung
Stationsausfall (_SC_STATION_DISCONNECTED (= 202))	16#39	16#F3	Der Empfänger des direkten Datenaustauschs empfängt keine Daten mehr.
Stationswiederkehr (_SC_STATION_RECONNECTED (= 203))	16#38	16#F0	Der Sender des direkten Datenaustauschs ist hoch gelaufen und sendefähig.
		16#F1	Der Empfänger des direkten Datenaustauschs ist hoch gelaufen und empfängt ohne Fehler bzw. Empfänger empfängt wieder Daten (alle Empfangsbereiche verfügbar).
		16#F2	Der Empfänger des direkten Datenaustauschs ist hoch gelaufen und empfängt mit Fehler bzw. Der Empfänger empfängt wieder Daten (mind. ein Empfangsbereich nicht verfügbar)

7.8.6 Alarme von SINAMICS S120 Antrieben

Beschreibung

Alarme, die vom SINAMICS S120 CU320/CBE20 bzw. SINAMICS S120 CU310 PN ausgelöst werden, werden über den PROFINET Alarmkanal abgesetzt. Bei den Alarmen sind zwei Typen zu unterscheiden:

- Alarme, die vom PROFINET Interface ausgelöst werden und im unmittelbaren Zusammenhang mit PROFINET stehen.
- Alarme, die von der Applikation/Technologie im Antrieb ausgelöst werden.

PROFINET-Alarme

Folgende Alarme werden über die PROFINET-fähige SINAMICS Baugruppe unterstützt:

Alarm	Beschreibung
Port Data Change Notification	Detaillierte Beschreibung finden Sie unter Alarme am IO-Controller (Seite 221)
Sync Data Changed Notification	
Isochronous Mode Problem Notification	
Multicast Communication Mismatch	

Alarme der Technologie/Applikation

Alarme werden nicht als Standard-PROFINET-Alarme an den Controller geschickt, sondern als PROFIdrive Alarm abgebildet. Diese müssen über einen Parameterauftrag abgeholt werden.

7.8.7 Systemfunktionen für die Diagnose für PROFINET bzw. PROFIBUS

Übersicht über System- und Diagnosefunktionen

Die folgende Tabelle zeigt eine Übersicht über die verschiedenen System- und Diagnosefunktionen für PROFINET IO. Unterschiede zu PROFIBUS DP werden ebenfalls aufgezeigt. Detaillierte Informationen zu den jeweiligen Funktionen finden Sie in den Referenzlisten der SIMOTION-Controller.

Funktion	Anmerkung	PROFIBUS	PROFINET
_getStateOfSingleDpSlave	Die Funktion gibt die Zustandsdaten eines DP-Slaves / IO-Devices zurück:	Logische Diagnoseadresse des DP-Slaves	Logische Diagnoseadresse des Stationsstellvertreters des IO Devices
_getStateOfAllDPStations	Die Funktion gibt die Zustandsdaten aller DP-Slaves / IO-Devices zurück:	Logische Diagnoseadresse des DP-Slaves	Logische Diagnoseadresse des Stationsstellvertreters des IO Devices
_activateDpSlave _deactivateDpSlave _getStateOfDpSlave	_getStateOfDpSlave liefert die Information, ob der Slave aktiviert oder deaktiviert ist.	Logischen Diagnoseadresse DP-Slave	Logische Diagnoseadresse des Stationsstellvertreters des IO Devices

Funktion	Anmerkung	PROFIBUS	PROFINET
_readDiagnosticData _getStateOfDiagnosticDataCommand	Realisiert das Auslesen der Diagnosedaten eines DP-Slaves durch das Anwenderprogramm. Die Diagnosedaten werden in der Form gelesen, wie sie durch EN 50170 Volume 2, PROFIBUS festgelegt sind. Struktur der Daten bei PROFINET ist nicht identisch zu PROFIBUS. Die Diagnose ist Subslot-granular	Logische Diagnoseadresse DP-Slave	Logische Diagnoseadresse oder IO-Adresse des Subslot
_readDriveFaults	Ermöglicht das Lesen des aktuellen Störpuffereintrages im Antrieb.	Logische Basisadresse des Antriebes (Slot)	Jede gültige logische I/O-Adresse des betreffenden Subslots oder Diagnose-adresse des PAP (für nutzdatenlose Subslots)

7.8.8 PROFINET Gerätediagnose in STEP 7

Gerätediagnose in STEP 7

Über HW Konfig kann im SCOUT eine ONLINE Gerätediagnose über PROFINET durchgeführt werden. Die Diagnose liefert neben Steckplatz, Kanalnummer auch die Fehlerart. Die Diagnose funktioniert analog zu PROFIBUS.

Vorgehensweise

1. Gehen Sie ONLINE und rufen Sie HW Konfig für das entsprechende SIMOTION-Gerät auf.
2. Wählen Sie **Zielsystem > Ethernet Teilnehmer diagnostizieren, beobachten/steuern** aus. HW Konfig sucht nach allen Netzwerkteilnehmern. Das Fenster **(Diagnose) ONLINE** wird eingeblendet und zeigt die Netzwerkteilnehmer an.
3. Klicken Sie mit der rechten Maustaste auf den gewünschten Teilnehmer und wählen Sie **Eigenschaften** aus. Die Detaildiagnose wird eingeblendet. Hier wird der entsprechende Fehler angezeigt.

Routing - Kommunikation über Netzwerksgrenzen 8

8.1 Was bedeutet Routing?

Routing bezeichnet das übermitteln von Informationen von einem Netz x in ein Netz y.

Man unterscheidet dabei grundlegend zwischen einem intelligenten, selbstlernendem Routing (z.B. IP-Routing im Internet) und einem Routing gemäß vorher festgelegten Routing-Tabellen (z.B. S7-Routing).

IP-Routing

Beim IP-Routing handelt es sich um ein selbstlernendes Routingverfahren (kann auch manuell durchgeführt werden) ausschließlich in Ethernet Kommunikationsnetzen die mit dem IP-Protokoll arbeiten, wie z.B. das Internet.

Die Funktion wird durch spezielle Router übernommen, die die Informationen anhand der IP-Adresse an benachbarte Netze weiterleiten, wenn die erkannte IP-Adresse im eigenen Netz nicht vorhanden ist.

S7-Routing

Beim S7-Routing handelt es sich um ein Routingverfahren, das auf vorher projektierten Routingtabellen aufbaut, dafür aber auch Informationen zwischen unterschiedlichen Kommunikationsnetzen austauschen kann, z.B. zwischen Ethernet, PROFIBUS und MPI. Diese Routingtabellen können als Verschaltungstabellen in NetPro angelegt werden.

Das S7-Routing arbeitet nicht mit der IP-Adresse, sondern mit so genannten Sub-Netz-IDs innerhalb des S7-Protokolls.

- Informationsübermittlung von Ethernet nach MPI und umgekehrt
- Informationsübermittlung von Ethernet nach PROFIBUS und umgekehrt
- Informationsübermittlung von MPI nach PROFIBUS und umgekehrt
- Informationsübermittlung von Ethernet nach Ethernet (nur SIMOTION, einschließlich PROFINET)

PG/PC- Zuordnung

Für S7-Routing ist eventuell eine Änderung der PG-Zuordnung nötig. Das können Sie jetzt in der Werkzeugleiste in SIMOTION SCOUT über den Button **PG zuordnen** durchführen. Damit rufen Sie das Eigenschaftsfenster zur PG-Zuordnung auf, in dem Sie die Zuordnung anpassen und "aktiv" (S7ONLINE-Zugriff) schalten können.

8.2 Projektierung von S7-Routing

Das S7-Routing wird im STEP7 / SIMOTION SCOUT mit Hilfe der Netzprojektierung "NetPro" projektiert.

Alle in der Netzprojektierung erfassten Stationen können Informationen untereinander austauschen. Dazu müssen Sie in NetPro Verbindungstabellen anlegen. Die erforderlichen Routingtabellen werden beim Übersetzen des Projektes automatisch erzeugt, müssen aber anschließend in alle beteiligten Stationen geladen werden.

8.3 Routing bei SIMOTION

Über Routing können Sie z.B. auf Geräte, die an Subnetze angeschlossen sind, über ein PG/PC ONLINE gehen.

IP-Routing

Das IP-Routing wird von SIMOTION grundsätzlich NICHT unterstützt, auch wenn sich auf manchen SIMOTION Geräten wie z.B. SIMOTION D4xx mehrere Ethernet-Schnittstellen befinden.

Falls Sie unterschiedliche Ethernet-Netze miteinander verbinden wollen, setzen Sie für das IP-Routing bitte einen separaten Router.

S7-Routing

Das S7-Routing wird von SIMOTION unterstützt, d.h. Informationen (Engineering-Zugriffe) können durch ein SIMOTION Gerät von überlagerten Netzwerken wie Ethernet und MPI an unterlagerte Netze wie PROFIBUS oder PROFINET/Ethernet (ab 4.1.2) durchgeroutet werden.

Randbedingungen

Beim Routen von Informationen auf einen taktsynchron betriebenen PROFIBUS sind folgende Randbedingungen in der Betriebsart "DP Slave" zu berücksichtigen:

- Die Funktionen "Äquidistanter Buszyklus" (Voraussetzung für taktsynchrone Anwendungen) und "Aktive Station" (Voraussetzung zum Routen ins unterlagerte Netzsegment) schließen sich gegenseitig aus.
- Der Betrieb aktiver I-Slave am taktsynchronen Bus ist nicht möglich.

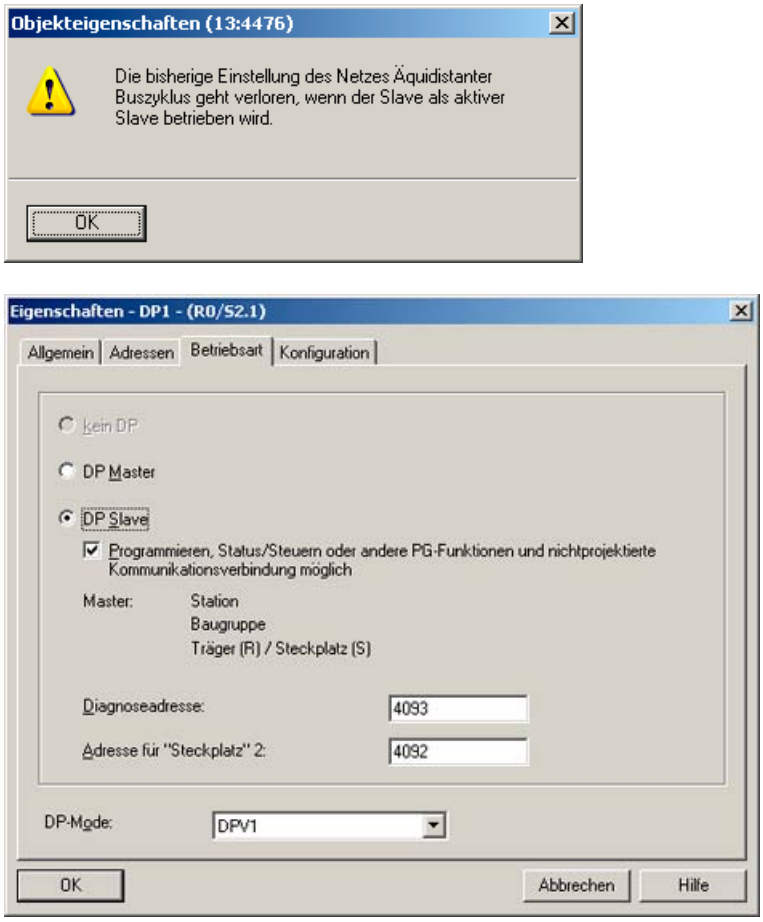


Bild 8-1 Betriebsart DP-Slave: Aktive Station: Test, Inbetriebnahme, Routing

Die Aktivierung des Kontrollkästchens "Programmieren, Status/Steuern oder andere PG-Funktionen …" muss gesetzt sein, wenn Sie z.B. PG-Funktionen, die bei der Inbetriebnahme und beim Testen benötigt werden, häufig über diese Schnittstelle durchführen wollen, oder wenn Sie mit PG-Funktionen (z.B. Starter) auf SINAMICS Antriebe an der kaskadierten, unterlagerten DP-Master Schnittstelle der SIMOTION durchgreifen (S7-Routen) wollen.

Das Aktivieren der Option " Programmieren, Status/Steuern oder andere PG-Funktionen …" bewirkt, dass die Schnittstelle zum aktiven Teilnehmer am PROFIBUS wird (d.h. die Schnittstelle ist am Token-Umlauf des Routing-PROFIBUS beteiligt). Folgende Funktionen sind dann möglich:

- Programmieren (z.B. laden)
- Testen (Status/Steuern)
- S7-Routing (I-Slave als Netzübergang)

Die Busumlaufzeit kann sich verlängern. Bei zeitkritischen Anwendungen, und wenn S7-Routing sowie die Client-Funktionalität für die Kommunikation nicht erforderlich sind, sollte diese Option deshalb nicht aktiviert werden.

Hinweis

Wenn das Kontrollkästchen "Programmieren, Status/Steuern oder andere PG-Funktionen..." nicht aktiviert ist, arbeitet die Schnittstelle nur als Server für Kommunikationsdienste, d.h. S7-Routing ist nicht möglich.

8.4 Routing bei SIMOTION D mit gestecktem PROFINET Board CBE30

Routing zwischen den verschiedenen Schnittstellen

Die beiden Standard Ethernetschnittstellen X120 bzw. X130 der SIMOTION D bilden je ein eigenes Subnetz, alle Ports auf CBE30 bilden ebenfalls ein gemeinsames Subnetz.

- Ein Routing von Subnetz zu Subnetz (IP-Routing) wird nicht unterstützt. Sie können dafür einen externen IP-Router einsetzen.
- Das S7-Routing von einem PROFINET/Ethernet-Subnetz zu einem PROFIBUS ist möglich.

Daraus leiten sich drei Möglichkeiten ab, ein PG/PC oder HMI über S7-Routing an eine SIMOTION D mit CBE30 anzuschließen.

Engineeringsystem an PROFINET (CBE30)

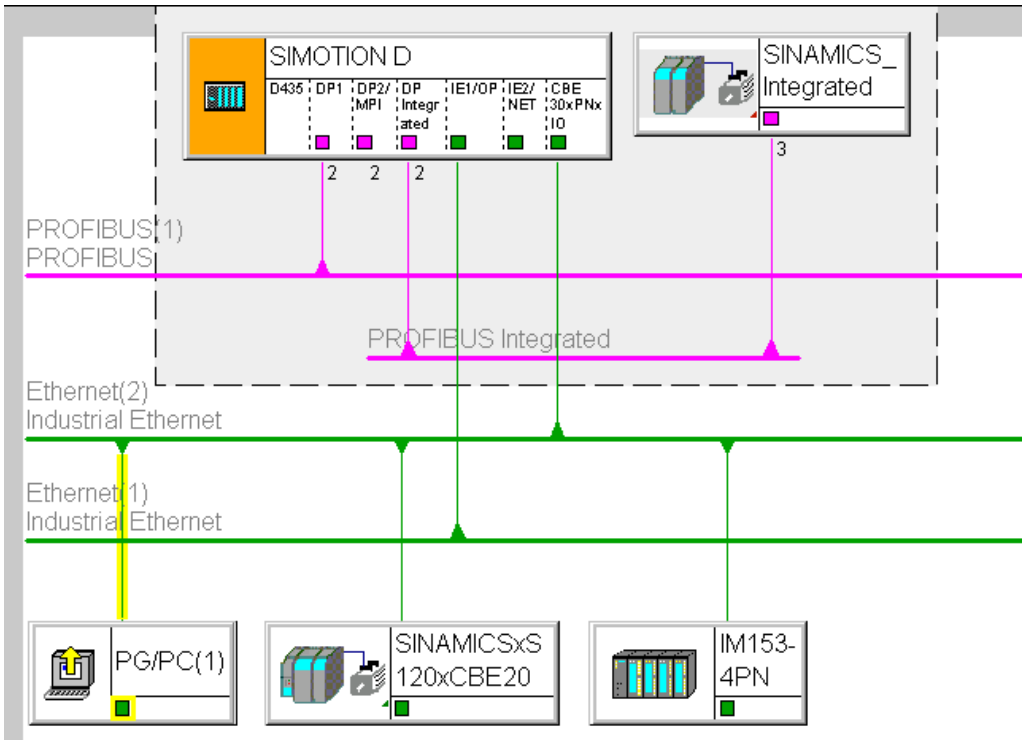


Bild 8-2 Beispiel für PG/PC an CBE30

- S7-Routing auf die (Master-) PROFIBUS-Schnittstellen (nur wenn projektiert)
- S7-Routing zum PROFIBUS Integrated
- S7-Routing zu den Standard-Ethernetschnittstellen ET1/ET2 (X120, X130) (ab V.4.1.2)
- Zugriff zu den Komponenten am gleichen Subnetz (CBE30) über die Switch-Funktionalität

Engineeringsystem/HMI an PROFIBUS

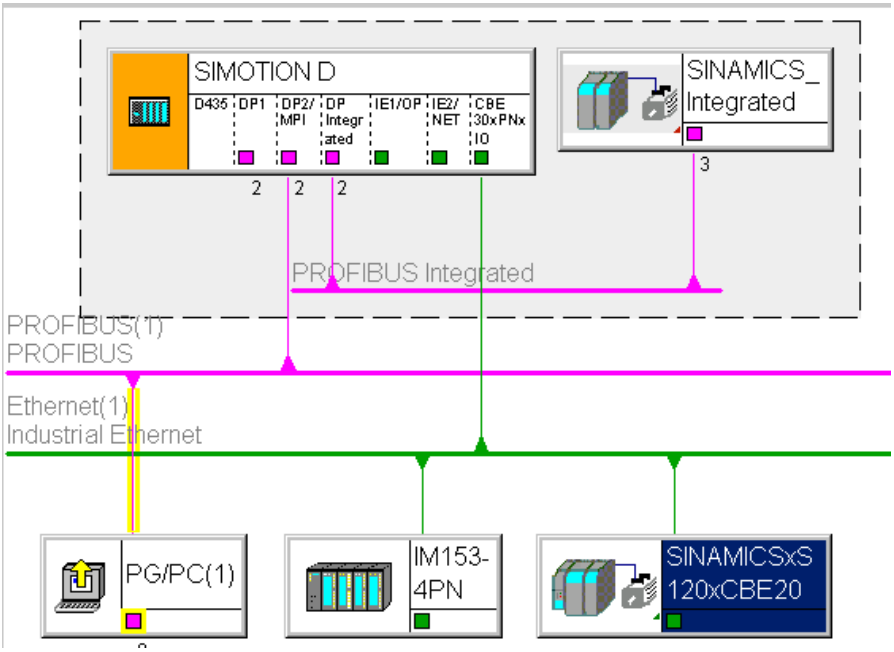


Bild 8-3 Beispiel für PG/PC an PROFIBUS

- S7-Routing zu den anderen (Master-) PROFIBUS- Schnittstellen (nur wenn projektiert)
- S7-Routing zum PROFIBUS Integrated
- S7-Routing zu X1400 auf dem CBE30
- S7-Routing zu den Standard-Ethernetschnittstellen (X120, X130) (ab V4.1.2)

Egineeringsystem/HMI an Ethernet

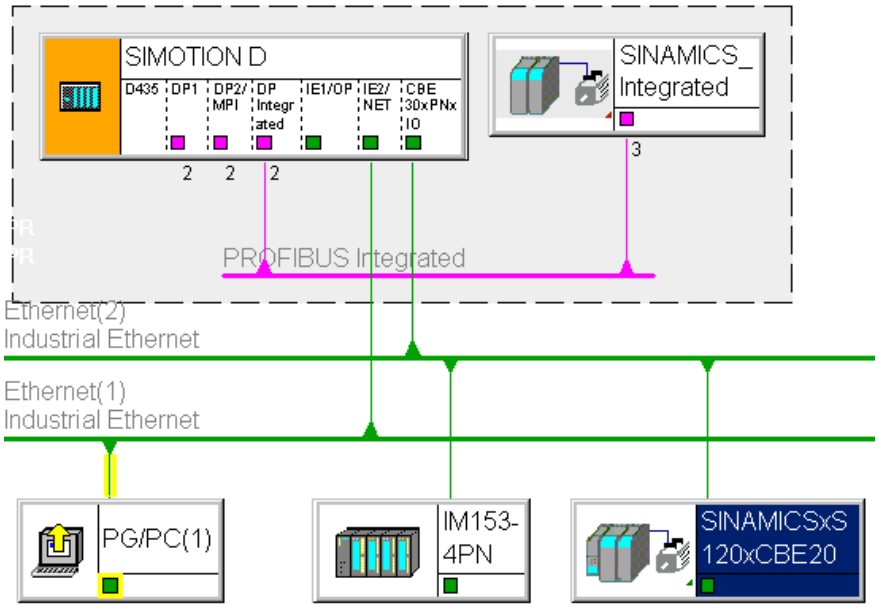


Bild 8-4 Beispiel für PG/PC an Ethernet X120, X130

- S7-Routing zu den anderen (Master-) PROFIBUS- Schnittstellen (nur wenn projektiert)
- S7-Routing zum PROFIBUS Integrated
- S7-Routing zu X1400 auf dem CBE30

8.5 Routing bei SIMOTION D zum SINAMICS integrated

S7-Routing zum internen PROFIBUS auf SINAMICS Integrated

Alle SIMOTION D haben eine SINAMICS Antriebsregelung integriert. Um auf Antriebs-Parameter zugreifen zu können, müssen die Telegramme von den externen SIMOTION D Schnittstellen auf den internen PROFIBUS DP durchgeroutet werden. Über S7-Routing können Sie auf den integrierten PROFIBUS zugreifen. Der interne PROFIBUS DP bildet dabei ein eigenes Subnetz. Dies ist insbesondere bei der Kommunikation auf mehrere Routing-Knoten zu beachten.

8.6 Routing bei SIMOTION P350

Beschreibung

S7-Routing ist möglich:

- von PROFIBUS (ISO-Board) auf PROFINET-Subnetz an MCI-PN-BOARD
- von PROFINET-Subnetz an MCI-PN-BOARD auf PROFIBUS (Iso-Profibus-Board)
- von Scout auf SIMOTION P über Softbus durch Run-Time hindurch auf PN-Geräte am MCI-PN-Board

Routing ist nicht möglich von Onboard-Ethernet-Schnittstellen auf PROFIBUS (Iso-Profibus-Board). IP-Routing ist über die Ethernet-Schnittstellen der P350 nicht möglich.

Routing von PROFIBUS auf PROFINET

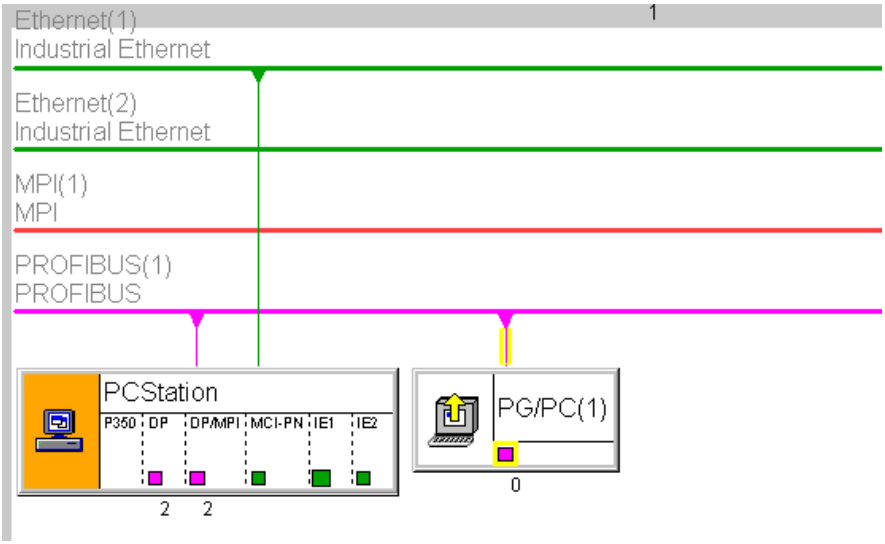


Bild 8-5 Beispiel für P350 Routing von PROFIBUS nach PROFINET

Routing von PROFINET auf PROFIBUS

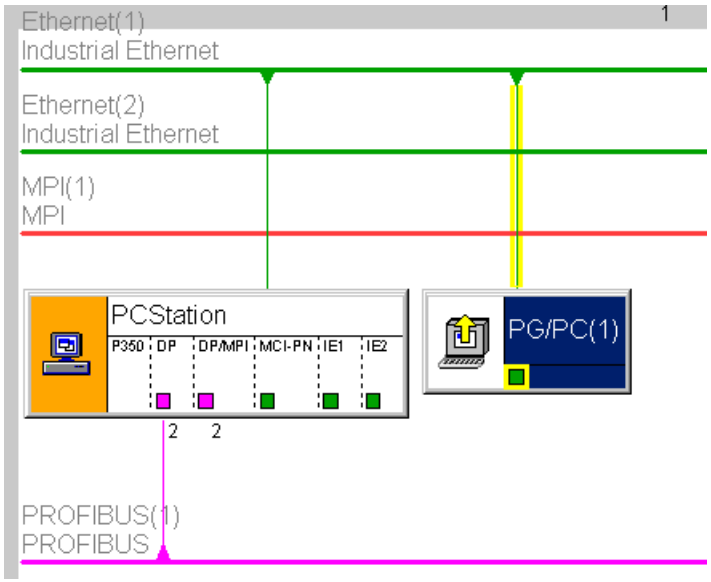


Bild 8-6 Beispiel für P350 Routing von PROFINET auf PROFIBUS

SIMOTION IT

9

9.1 SIMOTION IT - Übersicht

Beschreibung

SIMOTION IT bietet Ihnen die Möglichkeit mit Standard Internetmechanismen (HTTP) über Ethernet auf SIMOTION zuzugreifen und damit Diagnose und Prozessbeobachtung durchzuführen.

Dabei bieten sich folgende Vorteile.

- Ortsunabhängige offene Diagnose / Prozessbeobachtung
- Verwendung von Standardmechanismen (IP, TCP/IP, HTTP)
- Client-Gerät unabhängig vom Betriebssystem (Windows, Linux..)
- Unabhängig vom herstellerspez. Tool
- Entkoppelt vom Engineering
- Kein Versionskonflikt zwischen Client-Tool und Runtime
- Serieninbetriebnahme ohne Engineering-Tool

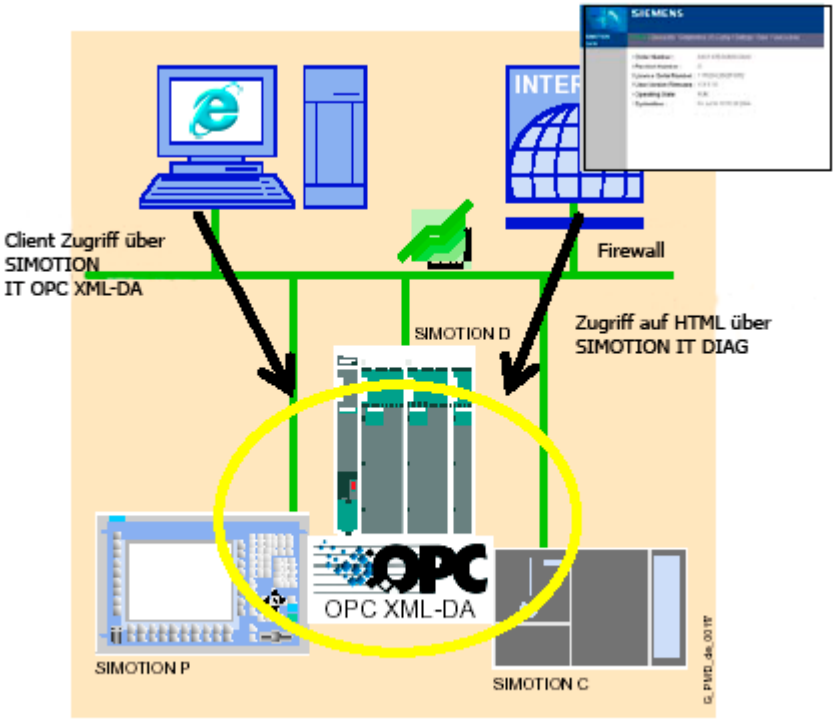


Bild 9-1 SIMOTION IT Übersicht

Dazu stellt SIMOTION verschieden Dienste zur Verfügung:

- SIMOTION IT DIAG
- SIMOTION IT OPC XML - DA
- Zugriff auf TRACE (Erweiterung von SIMOTION OPC XML - DA)
- Dateidownload über FTP (File Transfer Protokoll)

Weiterführende Literatur

Eine detaillierte Beschreibung der SIMOTION IT Produkte finden Sie in der Produktinformation **SIMOTION IT Ethernet basierende HMI- und Diagnosefunktionen** auf der CD SIMOTION SCOUT Dokumentation.

Siehe auch

Webzugriff auf SIMOTION (Seite 239)

SIMOTION IT DIAG (Seite 240)

SIMOTION IT OPC XML-DA (Seite 242)

9.2 Webzugriff auf SIMOTION

Beschreibung

Die nachfolgende Grafik zeigt die verschiedenen Möglichkeiten auf die Daten in einer SIMOTION Baugruppe zuzugreifen.

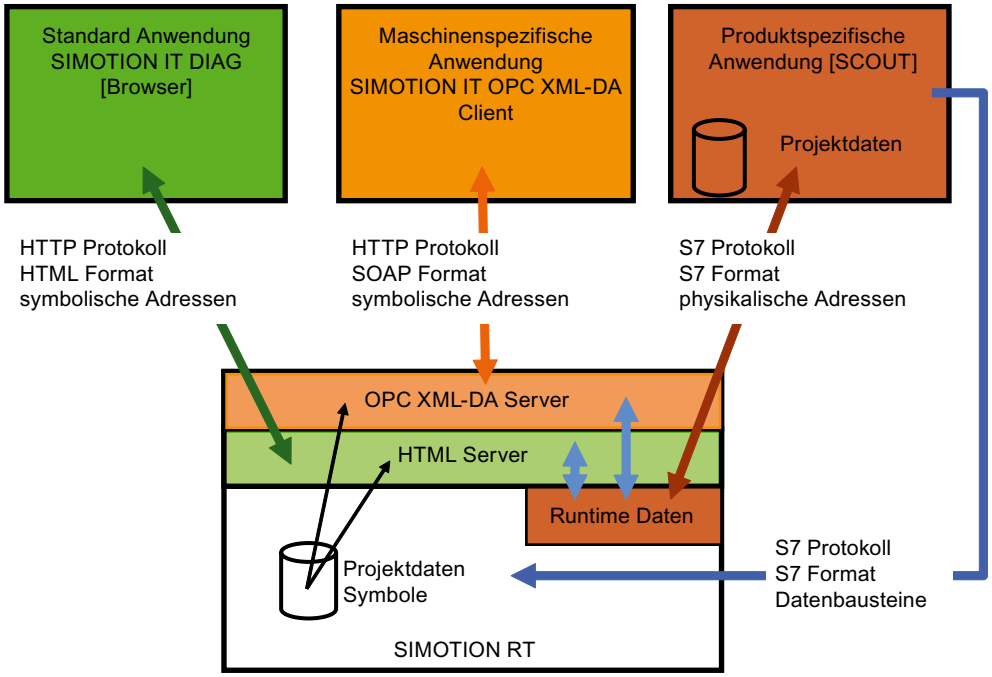


Bild 9-2 Zugriff auf SIMOTION

Siehe auch

SIMOTION IT DIAG (Seite 240)

SIMOTION IT OPC XML-DA (Seite 242)

9.3 SIMOTION IT DIAG

Beschreibung

SIMOTION IT DIAG bietet die Möglichkeit, von einem PC aus mit einem beliebigen Internetbrowser auf die HTML-Seiten in SIMOTION zuzugreifen.

Für SIMOTION IT DIAG ist eine eigene Lizenz erforderlich.

Standarddiagnoseseiten

SIMOTION bietet folgende Standarddiagnoseseiten:

- Startseite
- Device Info (Informationen über Firmware, Geräte, Gerätekomponenten und Technologieobjekte)
- Diagnostics (CPU-Auslastung, Speicherverbrauch, Betriebszustand, Tasklaufzeitenanzeige, Alarme SIMOTION und Drives, Trace, Diagnosepuffer, Serviceoverview.)
- IP-Config (Daten der Schnittstelle des SIMOTION-Gerätes)
- Settings (Einstellen der Zeitzone, Betriebszustände schalten, Anzeige benutzerdefinierten Seiten verändern)
- Manage Config (IT DIAG Konfigurationen laden, Deviceupdates, Daten vom Gerät speichern)
- Files (Zugriff auf das SIMOTION Filesystem, Up- und Download von Dateien, Erzeugen von Ordnern und Ablegen von weiteren Daten, z.B. Dokumentation

Vereinfachte Standardseiten

Für die optimierte Darstellung von IT DIAG Seiten auf Geräten, wie Handy oder PDA, wird ab der Version 4.1.3 ein Satz von speziellen Seiten bereitgestellt. Diese enthalten Informationen der Standardseiten in vereinfachter Darstellung. Einstiegspunkt für diese Seiten ist die Adresse http://<IPAddr>/BASIC.

Konfiguration über WebCfg.xml

Über die Konfigurationsdatei WEBCFG.XML wird der Webserver konfiguriert. Dies sind z. B. die Änderung von Standardseiten festlegen, Default-Seiten auf anwenderdefinierte Seite umstellen, Konfigurationsbereiche und Benutzerdatenbank festlegen.

Anwenderdefinierte Seiten

SIMOTION IT DIAG bietet die Möglichkeit individuell gestaltete Webseiten zu erstellen. Es stehen zwei verschiedene Varianten zur Auswahl:

- die Javascript Bibliotheken opcxml.js und appl.js
- die MiniWeb Server Language (MWSL)

Für die anwenderdefinierten HTML Seiten ist der Bereich "User's Area" in den Diagnose-Standardseiten vorgesehen. In diesen Bereich können Sie über das Flash Filesystem anwenderdefinierte HTML Seiten in der SIMOTION CPU ablegen.

Trace Interface

Das Funktionspaket "Trace Interface via SOAP" ermöglicht es, Variablenwerte in einen Puffer zu schreiben. Die Werte werden in Dateien gepackt und können per HTTP Request asynchron abgeholt werden. Die Nutzung dieser Schnittstelle erfordert grundsätzlich eine Applikation als Client. Der Client ermöglicht die Aufzeichnung eines zeitlichen Verlaufes von Variablen.

Variablenzugriff

Der Variablenzugriff ist für die SIMOTION IT Anwendungen über einen Variablenprovider realisiert. Dieser erlaubt es auf folgende Variablen zuzugreifen:

- Device-Systemvariablen
- TO-Systemvariablen
- Programm-Interface-Variablen
- Konfigurationsdaten
- Antriebsparameter
- Einstellen des Betriebszustands, Ausführen von RamToRom, Ausführen von ActiveToRom
- Technologische Alarme
- Diagnosepuffer

Sichere HTTPS Verbindung

Durch das Secure Socket Layer Protokol (SSL) wird eine verschlüsselte Datenübertragung zwischen einem Client und der SIMOTION ermöglicht. Das Secure Socket Layer Protokoll bildet die Basis für HTTPS Zugriffe des Browsers auf die SIMOTION CPU. Der verschlüsselte Zugriff auf eine SIMOTION kann sowohl über SIMOTION IT OPC XML-DA als auch über SIMOTION IT DIAG erfolgen.

9.4 SIMOTION IT OPC XML-DA

Beschreibung

Eine auf einem Client PC erstellte kundenspezifische Applikation, die z. B. mit der Programmiersprache C#, Visual Basic oder Java, programmiert ist, nutzt die Dienste und Eigenschaften von SIMOTION IT OPC XML-DA:

Für SIMOTION IT OPC XML-DA ist eine eigene Lizenz erforderlich.

- Offene Kommunikation über HTTP, SOAP, OPC-XML (Ethernet) zwischen Client – Gerät und SIMOTION (Webservices, Remote Procedure Call).
- Setzt auf die Spezifikation OPC XML DA 1.0 der OPC-Foundation auf
- Zugriff auf SIMOTION-Prozessvariable
  - Lesen/Schreiben von Variablen
  - Zyklische Lesen von Variablen über Subscriptions
  - Variablen browsen
- Trace Interface via SOAP; diese Funktion ist eine Erweiterung der OPC-Spezifikation
- Clients auf beliebiger HW mit unterschiedlichen Betriebssystemen (Windows, Linux, ..)
- Erstellung von Clientapplikationen mit C#, Java, C++. Die Applikation über die Sie auf den SIMOTION OPC-Server zugreifen möchten, müssen Sie selbst implementieren.
- Zugriffschutz über Benutzerkennung und Passwort

Die folgende Grafik zeigt schematisch den Zugriff auf den OPC-Server.

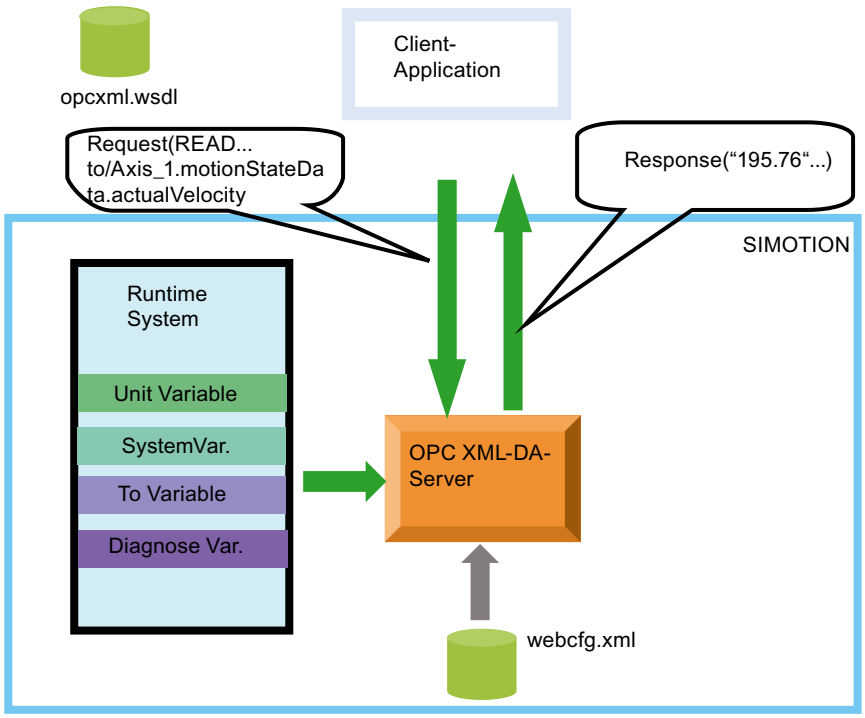


Bild 9-3 Zugriff auf den OPC-Server

C# Code-Beispiel

Tabelle 9- 1 Verbindungsaufbau

```
OpcXmlDa_R1_0.Service MyServer = new OpcXmlDa_R1_0.Service();
MyServer.Url = "http://" + this.ServerAddress.Text + "/soap/opcxml";
```

Tabelle 9- 2 Variable schreiben

```
WriteServer.Write(RequestOptions,WriteItemList,true, out RItemList,out
WriteErrorList);
```

Tabelle 9- 3 Subscription

```
OpcXmlDa_R1_0.ReplyBase Result = SubscribeServer.Subscribe(RequestOptions,
SubscribeItemList, false, 0, out SubscribeReplyItemList, out
SubscribeErrorList, out ServerSubHandle);
SubscribeServer.SubscriptionPolledRefresh(RequestOptions,
ServerSubHandleList,
Result.ReplyTime.AddMilliseconds(System.Double.Parse("100")),true,
System.Int32.Parse("10000"), false, out InvalidServerSubHandles, out
SubscribePolledRefreshReplyItemListArray, out SubscribeErrorList, out
DataBufferOverflow);
```

Variablenzugriff

Der Variablenzugriff ist für die SIMOTION IT Anwendungen über einen Variablenprovider realisiert. Dieser erlaubt es auf folgende Variablen zuzugreifen:

- Device-Systemvariablen
- TO-Systemvariablen
- Programm-Interface-Variablen
- Konfigurationsdaten
- Antriebsparameter
- Einstellen des Betriebszustands, Ausführen von RamToRom, Ausführen von ActiveToRom
- Technologische Alarme
- Diagnosepuffer

9.5 FTP Datentransfer

Filezugriff über FTP

Sie können gezielt auf die Daten der Speicherkarten von SIMOTION zugreifen. Dazu können Sie auf den in SIMOTION integrierten FTP-Server zugreifen. Sie müssen die Anwender und die entsprechenden Passwörter auf dem FTP-Server anlegen. FTP ist über einen Zugriffschutz verriegelt.

Über FTP können Sie z.B. Firmware Updates durchführen oder anwenderdefinierte HTML-Seiten laden.

Der FTP-Dienst erfordert keine eigene Lizenz.

Index

R

S

T

U

V

X